AN ABSTRACT OF THE DISSERTATION OF

Arash Shamaei for the degree of Doctor of Philosophy in Computer Science presented on December 10, 2015.

Title: Interconnection Networks Based on Gaussian and Eisenstein-Jacobi Integers

Abstract approved: _____

Bella Bose Mary Flahive

Quotient rings of Gaussian and Eisenstein-Jacobi(EJ) integers can be deployed to construct interconnection networks with good topological properties. In this thesis, we propose deadlock-free deterministic and partially adaptive routing algorithms for hexagonal networks, one special class of EJ networks. Then we discuss higher dimensional Gaussian networks as an alternative to classical multidimensional toroidal networks. For this topology, we explore many properties including distance distribution and the decomposition of higher dimensional Gaussian networks into Hamiltonian cycles. In addition, we propose some efficient communication algorithms for higher dimensional Gaussian networks including one-to-all broadcasting and shortest path routing. Simulation results show that the routing algorithm proposed for higher dimensional Gaussian networks outperforms the routing algorithm of the corresponding torus networks with approximately the same number of nodes. These simulation results are expected since higher dimensional Gaussian networks have a smaller diameter and a smaller average message latency as compared with toroidal networks.

Finally, we introduce a degree-three interconnection network obtained from

pruning a Gaussian network. This network shows possible performance improvement over other degree-three networks since it has a smaller diameter compared to other degree-three networks. Many topological properties of degree-three pruned Gaussian network are explored. In addition, an optimal shortest path routing algorithm and a one-to-all broadcasting algorithm are given.

Interconnection Networks Based on Gaussian and Eisenstein-Jacobi Integers

by

Arash Shamaei

A DISSERTATION

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Doctor of Philosophy

Presented December 10, 2015
Commencement June 2016

Doctor of Philosophy dissertation of Arash Shamaei presented on December 10, 2015.

APPROVED:

Co-Major Professor, representing Computer Science

Co-Major Professor, representing Computer Science

Director of the School of Electrical Engineering and Computer Science

---

Dean of the Graduate School

I understand that my dissertation will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my dissertation to any reader upon request.

Arash Shamaei, Author

ACKNOWLEDGEMENTS

I wish to thank my dear wife whose love and support inspired me to complete this thesis.

I would like to express my sincere appreciation and gratitude to my advisor Professor Bella Bose, and my co-advisor Professor Mary Flahive for their continuous support during my Ph.D study. Their knowledge, patience, guidance, and support helped me to finish this journey successfully.

Special thanks go to my committee, Dr. Ben Lee , Dr. Lizhong Chen , and Dr. Ross Hatton for their valuable guidance and suggestions.

Finally I wish to thank all the faculty and the staff who supported me during my study in the school of Electrical Engineering and Computer Science at Oregon State University.

This research was supported by the National Science Foundation grants CCF-1015804 and CCF-1423656.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

LIST OF TABLES

Chapter 1: Introduction

Supercomputers and parallel computers consist of many individual nodes connected by edges in an interconnection network. The nodes communicate with each other by passing messages along the edges of the network using a standard message passing mechanism such as the Message Passing Interface (MPI) [10]. The topology chosen for the interconnection network therefore plays an integral role in the performance of the computer. One of the more popular choices is the toroidal topology, which includes toroidal meshes and the k-ary n-cube. For example, the IBM BlueGene/Q [10], the Cray XE6 and XK7 (3D torus gemini interconnect [3]), the HP GS1280 multiprocessor [14], the K-computer [1], and the Intel Ivy bridge [18] all use a toroidal topology.

Eisenstein-Jacobi networks (often abbreviated as EJ networks) are an alternative to 2D tori. Originally developed over two decades ago [30], an EJ network is a 2D wraparound network of degree six. Many topological properties of these networks were explored in [37], [24]. It is shown in [2] that EJ networks are a generalization of the hexagonal torus networks developed earlier in [31], [11], and [22]. In particular, the hexagonal torus topology was used in the design of the Hexagonal Architecture for Real-Time Systems (HARTS) machine at the University of Michigan [31]. The Mayfly [19] designed at HP laboratories is another example of a parallel system that uses a hexagonal torus topology and a hexagonal topology has been proposed for cellular networks [27]. In Chapter 2 we develop two deadlock-free routing algorithms for hexagonal torus interconnection networks.

Gaussian networks [38] were proposed as another alternative to toroidal networks. Both a 2D torus and a Gaussian network are of degree four whereas the latter has smaller diameter than a torus with an approximately the same number of nodes. A smaller diameter directly translates to a smaller average message latency, the average time of arrival and departure of messages in the network. Su-

percomputers with large number of nodes usually employ n-dimensional tori. For example, the IBM BlueGene/Q [10] uses a 5D torus. Gaussian networks are 2D. In Chapter 3 we develop higher dimensional Gaussian networks as an alternative to higher dimensional tori and many topological properties of higher dimensional Gaussian networks are proved.

Degree-three networks are known to cost less than 2D toroidal networks where the cost of a network is defined as the product of the degree and the diameter of the network [44]. One well-known class of degree-three networks contains the honeycomb networks introduced in [44]. Some honeycomb networks are shown to have similar properties compared to 2D tori. It is shown in [41] that a honeycomb torus can be obtained by pruning a 2D torus, and in Chapter 4 we apply a similar pruning technique to Gaussian networks and construct degree-three pruned Gaussian networks. Just as Gaussian networks are a generalization of 2D squared torus networks, we show that degree-three pruned Gaussian networks are a generalization of honeycomb squared torus networks. The resulting degree-three pruned Gaussian network has a smaller diameter than the corresponding honeycomb torus with an equivalent number of nodes.

The rest of this thesis is organized as follows. Chapter 2 contains two deadlock-free routing algorithms for hexagonal meshes and tori. Higher dimensional Gaussian networks along with their topological properties are given in Chapter 3. Chapter 4 describes degree-three pruned Gaussian networks along with some topological properties and communication algorithms. Finally, this thesis ends with some concluding remark in Chapter 5.

Chapter 2: Deadlock-free Routing in Hexagonal Interconnection Networks

2.1 Introduction

The hexagonal torus topology was used in the design of the Hexagonal Architecture for Real-Time Systems (HARTS) machine at the University of Michigan [31]. The Mayfly [19] designed at HP laboratories is another example of a parallel system that uses a hexagonal torus topology. In addition, the hexagonal topology has been proposed for cellular networks [27].

Shortest path routing algorithms for hexagonal tori were originally proposed in [31], [11], and [22]. These original algorithms used a three-component addressing scheme based on the decomposition of the network into three edge-disjoint Hamiltonian cycles. Each of the components specifies the order of a node in one of the three Hamiltonian cycles. This was improved in [27] to require fewer bits in each component. Here the node addresses (x, y, z) represent the corresponding position along the angles 0, 60 and 120 degrees from the horizontal axis. It is shown in [27] that under this representation, a node can be represented in more than one possible way and so the representation has some redundancy. Later on, an improved two-component addressing scheme was given [37] and [2]. Under this addressing scheme, each component of the address represents the position of a the node at angles of 0 and 60 degrees from the horizontal axis. (This is summarized in Section 2.2.) This scheme was proved in [2] to have a minimal number of addressing bits.

Deadlock can be informally described as follow. Consider a group of agents - which are usually packets in our discussion - that are each supposed to travel along the network from a source to a destination. There might be a situation in which these agents cannot proceed and must wait forever in the network. This

situation happens when some group of agents are waiting for each other to release resources. A resource can be either a physical channel, or a virtual channel, or a buffer. When a sequence of waiting agents forms a cycle then deadlock is said to occur in the network [17, Chapter 14].

The authors in [32] provide a minimum fraction of turns that must be prohibited in hexagonal meshes and hexagonal tori (as well as in some other regular topologies) in order to construct a deadlock-free routing algorithm in an interconnection network. Prohibited turns are the turns that are not allowed to be taken by a routing algorithm.

The *resource dependence graph* can be defined as a directed graph $G = (V, E)$. The set of nodes can be resources or agents. If an agent A is holding a resource node B then there is an edge from A to B. In addition, if an agent A is waiting for a resource B then there is an edge from B to A. In this resource dependence graph, if there is a cycle then there is a possibility of deadlock in the network [15].

Cycles in the resource dependence graph are necessary but not sufficient conditions for deadlock. Dally and Seitz [15] showed deadlock can be avoided if there is a total ordering on the resources such that every packet is routed in increasing (or decreasing) order of resources. This condition is too restrictive and it is not necessarily required as pointed out in [23].

In this chapter we first propose a deadlock-free minimal routing algorithm for hexagonal mesh networks based on deadlock avoidance techniques. Here, the term minimal means that at every step in the routing algorithm the packet gets closer to its destination. The algorithm is a deterministic routing; that is, there is exactly one path that any packet can travel from source to destination.

Next we extend the proposed algorithm to hexagonal torus networks by adding some extra virtual channels to the network in order to avoid the deadlock caused by the wraparound edges. In order to verify the proposed algorithm is deadlock-free, we use Dally and Seitz's technique [15] of establishing a total ordering on the resources. Two virtual channels for the deadlock-free deterministic routing algorithm are used. How the virtual channels are used is similar to the routing

given in [46] for toroidal networks.

Next we propose a partially adaptive minimal routing for hexagonal mesh networks. A partially adaptive routing algorithm offers more than one minimal path for some source and destination pairs. This algorithm applies Glass and Ni's turn model [28] to hexagonal mesh networks. The basis for our algorithm is shortest path routing [2], modified by removing cycles from the resource dependence graph.

Finally, we extend the partially adaptive routing algorithm to hexagonal torus networks. This time the algorithm uses three virtual channels to remain deadlock-free.

The rest of the chapter is organized as follows. Section 2.2 briefly explains the topology of hexagonal torus networks and the turn model is reviewed in Section 2.3. A deterministic and deadlock-free routing algorithm for hexagonal meshes and hexagonal tori are presented in Section 2.4. Section 2.5 shows a partially adaptive and deadlock-free routing algorithm for hexagonal meshes and hexagonal tori. The summarizing conclusions are given in Section 2.6.

2.2 The Hexagonal Torus Topology

As discussed earlier, the hexagonal torus topology was used in the design of HARTS network at the University of Michigan [31], [11], [22]. For every positive integer n, the hexagonal network is denoted by $\mathcal{H}_n$ and its nodes are the points in the infinite equilateral triangle grid that lie on the first n concentric hexagons about the origin. Each node of the infinite equilateral grid lies on six edges, labeled according to its direction by E, NE, NW, W, SW, or SE. It is shown in [24] that every node in this infinite grid can be addressed by a two-component addressing scheme as $\langle x, y\rangle$ where x and y correspond to the signed distance of E and NE from the origin, respectively. For instance, the nodes on the ray at an angle of 60 degrees through the origin have coordinates $\langle 0, n\rangle$ for all $n = 0, 1, \ldots$ while those on the ray at an angle of $-\pi/3$ have coordinates $\langle n, -n\rangle$ for all $n = 0, 1, \ldots$. Fig. 2.1 gives the addresses for the nodes of $\mathcal{H}_3$.

The network $\mathcal{H}_n$ has two types of edges: *regular* and *wraparound* edges. Edges

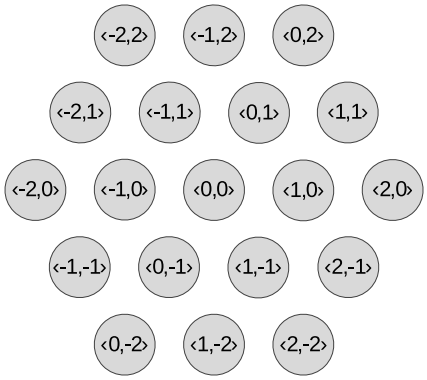


Figure 2.1: The node addresses for $\mathcal{H}_3$.

of the first type are the sides of all triangles in the infinite triangle grid that connect two nodes of $\mathcal{H}_n$. The second type consists of the wraparound edges, connecting two boundary nodes in $\mathcal{H}_n$. In order to explain the wraparound edges, the concept of EJ numbers and the quotient rings are useful. These concepts are briefly explained next.

Let ω be the complex number $\frac{1+i\sqrt{3}}{2}$, defining the NE direction in the infinite grid. Every node $\langle x, y\rangle$ in the infinite triangle grid can be uniquely represented by the complex number $x + y\omega$ where x and y are integers. These complex numbers are called the *Eisenstein-Jacobi* numbers or EJ numbers and they form a subring of the field of complex numbers. An EJ interconnection network [30], [37] is generated by a fixed EJ number $\alpha = a + b\omega$. Its nodes are the congruence classes modulo α where the nodes $x = x_1 + x_2\omega$ and $y = y_1 + y_2\omega$ are adjacent if and only if $y - x$ equals ± 1, $\pm\omega$, or $\pm\omega^2$ modulo α. The EJ network generated by α is homogeneous, regular of degree six, and always has $|\alpha|^2 = a^2 + ab + b^2$ nodes.

The hexagonal network $\mathcal{H}_n$ is the EJ network generated by $\alpha = n + (n-1)\omega$, with $|\alpha|^2 = 3n^2 - 3n + 1$ nodes. The diameter of the hexagonal network $\mathcal{H}_n$ is $n - 1$ and has the maximal number of nodes for all regular networks of degree 6 and diameter $n - 1$ [24].

Now the wraparound edges are explained. An example is considered first. Consider $\mathcal{H}_3$. First tile $\mathcal{H}_3$, which is generated by $\alpha = 3 + 2\omega$ in the infinite grid,

with the centers at $(s + t\omega)\alpha$, where s and t are integers. Fig. 2.2 shows three of these tiles.

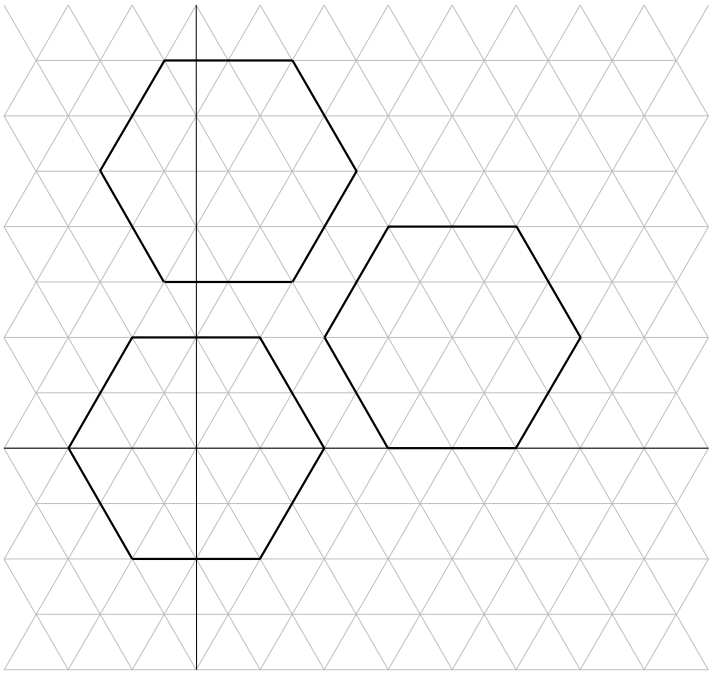


Figure 2.2: Three hexagons in the tiling of the plane by hexagons where the centers of the hexagons are $0 = \langle 0, 0\rangle$, $3 + 2\omega = \langle 3, 2\rangle$, and $2\omega^2 + 3\omega = \langle -2, 5\rangle$

The nodes of a hexagonal network lie within a hexagon centered at the origin. In the infinite triangle grid every node has six neighbors. For instance, the node $\langle 0, 2\rangle$ is adjacent to six nodes: $\langle -1, 3\rangle$, $\langle 0, 3\rangle$, $\langle 1, 2\rangle$, $\langle 1, 1\rangle$, $\langle 0, 1\rangle$, and $\langle -1, 2\rangle$ (listing clockwise). The last three neighbors are in $\mathcal{H}_3$, but the first three neighbors are not. The two nodes $\langle -1, 3\rangle$ and $\langle 0, 3\rangle$ are located in the hexagon centered at $\langle -2, 5\rangle$ and the node $\langle 1, 2\rangle$ is in hexagon centered at $\langle 3, 2\rangle$. When these hexagons are translated to be centered at the origin, these points correspond to $\langle 1, -2\rangle$, $\langle 2, -2\rangle$, and $\langle -2, 0\rangle$, respectively. This is equivalent to find mod $\alpha = 3 + 2\omega$ for the corresponding nodes. In $\mathcal{H}_3$, the node $\langle 0, 2\rangle$ is adjacent (by wraparound edges) to the nodes $\langle 1, -2\rangle$, $\langle 2, -2\rangle$, and $\langle -2, 0\rangle$.

In general the wraparound edges are obtained by considering the tiling of the

infinite triangle grid with non-overlapping hexagons. Consider H_n which is generated by $\alpha = n + (n-1)\omega$ and centered at the origin. Next tile the infinite triangle grid with non-overlapping hexagons with centers at $(s + t\omega)\alpha$ for $s, t \in \mathbb{Z}$. Each node in H_n has six neighbors in the six different directions. (that is, in $\omega^0, \ldots, \omega^5$ directions.) Consider the boundary nodes. These nodes have some neighbors which are not located within H_n but are located in other hexagons centered at $\omega^j\alpha$ for $j = 0, \ldots, 5$ where $\omega^j\alpha$ is of the form $(s + t\omega)\alpha$. The neighbors of the boundary nodes are obtained by translating back the hexagons centered at $\omega^j\alpha$ to be centered at the origin. Fig 2.3 shows an example of wraparound edges for H_4.

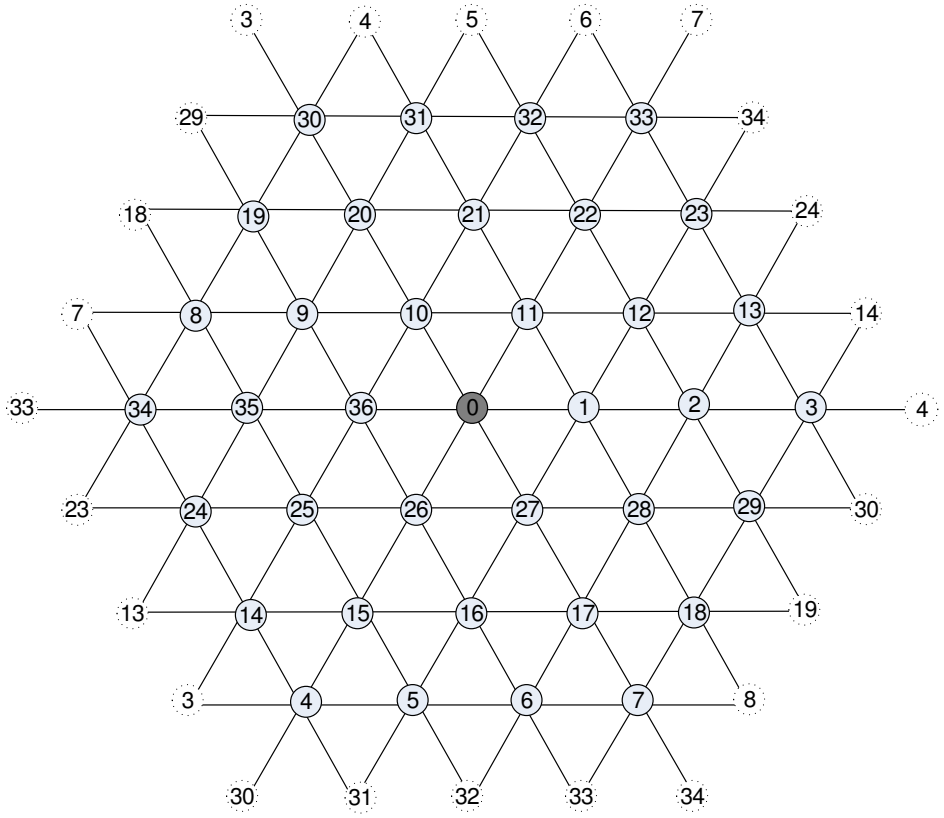


Figure 2.3: The wraparound edges for H_4. Shaded nodes show the networks nodes. For simplicity, the nodes are numbered in order they are visited when traveling along the Hamiltonian cycle in the East direction starting from node 0.

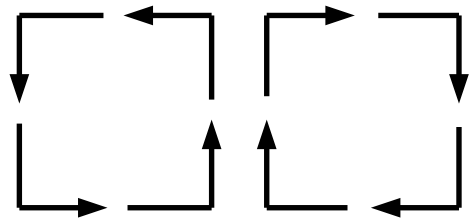
2.3 The Turn Model

The turn model was introduced by Glass and Ni in [28] as a technique to prevent deadlock in wormhole switched interconnection networks. Unlike other deadlock avoidance methods, the turn model does not add virtual channels to the network but rather eliminates the possibility of cycles in the resource dependence graph by forbidding some turns. It was originally described for $2D$ mesh networks and extended to $2D$ torus networks. The turn model is deadlock free, livelock free, and maximally adaptive [28]. We will apply the turn model to the hexagonal network to avoid deadlock in routing. First we briefly review the turn model for the $2D$ mesh.

Consider a 2D mesh with sides parallel to the X-axis and Y-axis where x and y represent the two dimensions. Packets traveling in one dimension can cause dependency between channels in the same dimension (and direction) and channels of the next dimension. These dependencies are drawn in the form of eight possible turns from one dimension to the other dimension as shown in Fig. 2.4. For example, the bottom-right turn in Fig. 2.4(a) represents a packet traveling in the $+x$ direction and this packet can wait on the channel in the $+y$ direction. Allowing all eight turns generates cycles in the resource dependence graph, and one way to avoid deadlock is to remove some turns so that there is no cycle in the resource dependence graph [15]. The well-known dimension order routing (first going along the X-dimension and then along the Y-dimension) forbids the four turns (two from each cycle) given in Fig. 2.5. Because of this, the dimension order routing is deadlock-free [28].

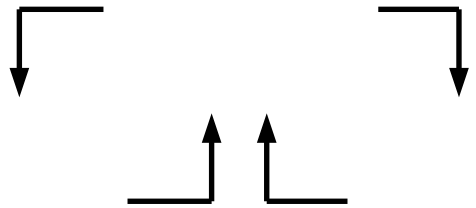
Instead of avoiding these turns, the turn model proposed in [28] avoids only two turns, one from each cycle, and so it is more adaptive than the dimension order routing. In each of the two cycles in Fig. 2.4, any one of the four turns can be removed to avoid cycles and so there are a total of 16 possible ways of removing one turn from each cycle.

Of the 16 possible ways to remove one turn from each cycle, it is known that twelve of them avoid cycles [28]. For instance it is impossible to generate a cycle



(a) Counter clockwise turns (b) Clockwise turns

Figure 2.4: Eight possible turns in a $2D$ mesh.



(a) Counter clockwise turns (b) Clockwise turns

Figure 2.5: Four turns are eliminated in dimension order routing.

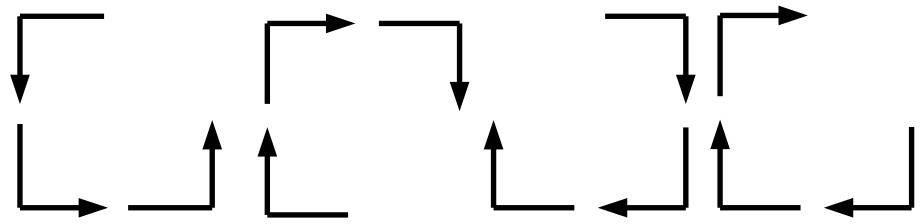
with any combination of the turns in Fig. 2.6(a) with those in Fig. 2.6(b) [28]. The routing algorithms resulting from these combinations are named (in order): the west-first, the north-last, and the negative-first.

In the next section, similar concepts are used to avoid deadlock for minimal routings in hexagonal networks.

2.4 Deterministic Routing in Hexagonal Mesh and Torus Networks

Each node in a hexagonal mesh network has a two-component address $\langle x, y \rangle$ where x, y are integers and $x + y\omega$ is its representation in the infinite triangle grid [2]. Note that the complex number $\omega = \frac{1+i\sqrt{3}}{2}$ and

$$\omega^2 = \omega - 1 \ ; \ \omega^3 = -1 \ ; \ \omega^4 = -\omega \ ; \ \omega^5 = 1 - \omega \ ; \ \omega^6 = 1.$$



(a) Forbidden turn (b) West-first, North-last, and Negative-first

Figure 2.6: The turn model for $2D$ mesh. (a)The North to West turn is not allowed. (b)Three possible routing algorithms.

Type	Signs	Route
1	$x, y \geq 0$	x nodes in ω^0, y nodes in ω^1
2	$x \leq 0, y \geq 0, \lvert x\rvert \leq \lvert y\rvert$	$\lvert y\rvert - \lvert x\rvert$ nodes in ω^1, $\lvert x\rvert$ nodes in ω^2
3	$x \leq 0, y \geq 0, \lvert x\rvert \geq \lvert y\rvert$	y nodes in ω^2, $\lvert x\rvert - \lvert y\rvert$ nodes in ω^3
4	$x, y \leq 0$	$\lvert x\rvert$ nodes in ω^3, $\lvert y\rvert$ nodes in ω^4
5	$x \geq 0, y \leq 0, \lvert x\rvert \leq \lvert y\rvert$	$\lvert y\rvert - \lvert x\rvert$ nodes in ω^4, x nodes in ω^5
6	$x \geq 0, y \leq 0, \lvert x\rvert \geq \lvert y\rvert$	$\lvert y\rvert$ nodes in ω^5, $\lvert x\rvert - \lvert y\rvert$ nodes in ω^0

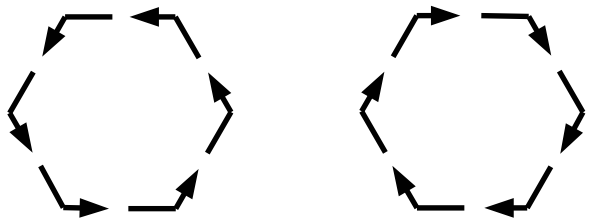
Table 2.1: Shortest path routing from origin to node $\langle x, y\rangle = a\omega^{j-1} + b\omega^j$.

For every grid point $\langle x, y\rangle$ there exists j such that it can be written as $a\omega^{j-1} + b\omega^j$ where a and b are *nonnegative* integers (this is proved in [24]). The sign and comparative size of x and y determines j. Using this representation, in [24] it is proved that the minimal distance from the origin to the grid point $a\omega^{j-1} + b\omega^j$ is $a + b$ and the route from origin to this point is obtained by taking any a edges of direction ω^{j-1} and b edges of direction ω^j. The message taking this route is referred as *type-j* message ($j = 1, 2 \ldots 6$). Table 2.1 summarizes the six different message types and their route in hexagonal networks.

In general, to route from S to D first $D - S = x + y\omega \mod \alpha$ is calculated. Then $D - S$ is written as $a\omega^{j-1} + b\omega^j$ for integer $a, b \geq 0$ according to Table 2.1. The routing is done by taking any a edges of direction ω^{j-1} and b edges of direction

ω^j from the source node [24].

Note that in a type-j message there might be a 120 degree counter-clockwise turn from ω^{j-1} to ω^j. Fig. 2.7(a) illustrates how these turns can create cycles in the resource dependence graph which can result in a deadlock situation. Similarly, a type-j message may contain a 120 degree clockwise turn from ω^j to ω^{j-1}. Fig. 2.7(b) shows how these turns can also generate a cycle.



(a) Counter-clockwise turns (b) Clockwise turns

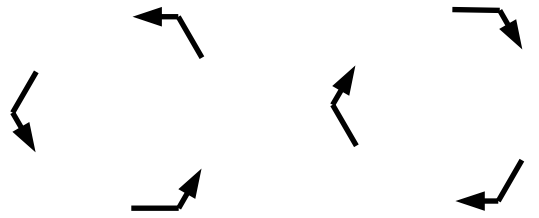
Figure 2.7: 12 possible turns in an hexagonal network

Looking at the infinite triangle grid one asks what kinds of cycles might occur. Some convex polygons including triangle, rhombus, parallelogram, and isosceles trapezoid might create cycles because these polygons have at least one 60 degree interior angle. A 60 degree turn consists of a turn from ω^j direction to $\omega^{j\pm 2 \mod 6}$ direction. Such a turn is not permitted by the shortest path routing algorithm. Therefore no cyclic channel dependency of the shape of the mentioned polygons can be constructed.

2.4.1 A Deterministic Algorithm in Hexagonal Mesh Networks

Suppose S and D are the source and destination nodes respectively. From the representation $D - S = a\omega^{j-1} + b\omega^j$ where $a, b \geq 0$ it is known that every minimal path uses any combination of a edges in the ω^{j-1} direction and b edges in the ω^j direction. Our deterministic algorithm will choose only one of these paths. Showing that our algorithm avoids the cycles shown in Fig. 2.7 will guarantee that it is deadlock-free.

Of the twelve possible turns, only the six turns shown in Fig. 2.8 will be allowed. Given $D - S = a\omega^{j-1} + b\omega^{j}$, our algorithm chooses the following minimal path: If j is one of $j = 1, 3, 5$, travel a edges in the ω^{j-1} direction followed by b edges in the ω^{j} direction. For $j = 2, 4, 6$ first travel b edges in the ω^{j} direction and then a edges in the ω^{j-1} direction. This algorithm only uses the six turns in Fig. 2.8.



(a) Counter clockwise turns (b) Clockwise turns

Figure 2.8: Six of the twelve turns are allowed in deterministic routing.

In order to prove that this algorithm is deadlock-free, we assign a numbering system to the channels that is similar to the one given in [15]. In the algorithm, a packet traveling in the ω^0 direction can only wait for a channel in the ω^0, ω^1 or ω^5 directions. Similarly, an ω^2 packet can waits only for the ω^2, ω^1 or ω^3 directions, and an ω^4 packet waits only for the ω^4, ω^3 or ω^5 directions. We will show that this allows the channels to be numbered in a way such that the sequence of channels traveled by any packet is in strictly increasing or strictly decreasing order. This guarantees the algorithm is deadlock-free [15]. As mentioned earlier, this approach is a sufficient condition and it is too restrictive as stated in [23].

It is shown in [24] that there exist three edge-disjoint Hamiltonian cycles in the EJ network generated by $\alpha = a + b\omega$ provided that $\gcd(a, b) = 1$. These Hamiltonian cycles are obtained by successively adding $1, \omega$ or ω^2 modulo α to the origin. Note that a hexagonal torus network is generated by $\alpha = n + (n-1)\omega$ and $\gcd(n, n-1) = 1$. Therefore, there exist three edge-disjoint Hamiltonian cycles in a hexagonal torus network.

One way to assign a number to the channels is to use the order in which the channels are visited along one Hamiltonian cycle. In this subsection we consider

hexagonal meshes and the Hamiltonian cycles explained above do not exist in hexagonal meshes. For simplicity, we imagine hexagonal meshes as hexagonal tori (that is, with wraparound channels). As we visit the channels along a Hamiltonian cycle when we encounter wraparounds no number is assigned.

The channels are numbered as follows. (Fig. 2.9 gives the numbering for $\mathcal{H}_3$). First we assign numbers to the channels that belong to ω^0, ω^2 and ω^4 directions and then to the channels that belong to ω^1, ω^3 and ω^5 directions. The numbers are assigned in the increasing order determined by the Hamiltonian cycle in each direction. The numbering at each direction starts from the given node and ends at the original node along the corresponding Hamiltonian cycle.

- ω^0-direction: Start numbering at node $-(n-1)$.
- ω^2-direction: Continue numbering at node $(n-1)-(n-1)\omega$.
- ω^4-direction: Continue numbering at node $(n-1)\omega$.
- ω^1-direction: Continue numbering at node $-(n-1)\omega$.
- ω^3-direction: Continue numbering at node $(n-1)$.
- ω^5-direction: Continue numbering at node $-(n-1)+(n-1)\omega$.

Theorem 2.1. *The deterministic routing algorithm for hexagonal meshes defined by the turns in Fig. 2.8 is deadlock-free.*

Proof. Since in each direction the assigned numbers of the channels increase along that direction, it suffices to prove the values of channels in an allowed turn increase along that turn. Each allowed turn in the algorithm originates in one of the directions $\omega^0, \omega^2, \omega^4$ and ends in one of the directions $\omega^1, \omega^3, \omega^5$. Since all numbers of the channels in the first group of directions are less than those in the second group of directions, the values of the two channels in an allowed turn are increasing along the turn. This guarantees that the algorithm is deadlock-free. □

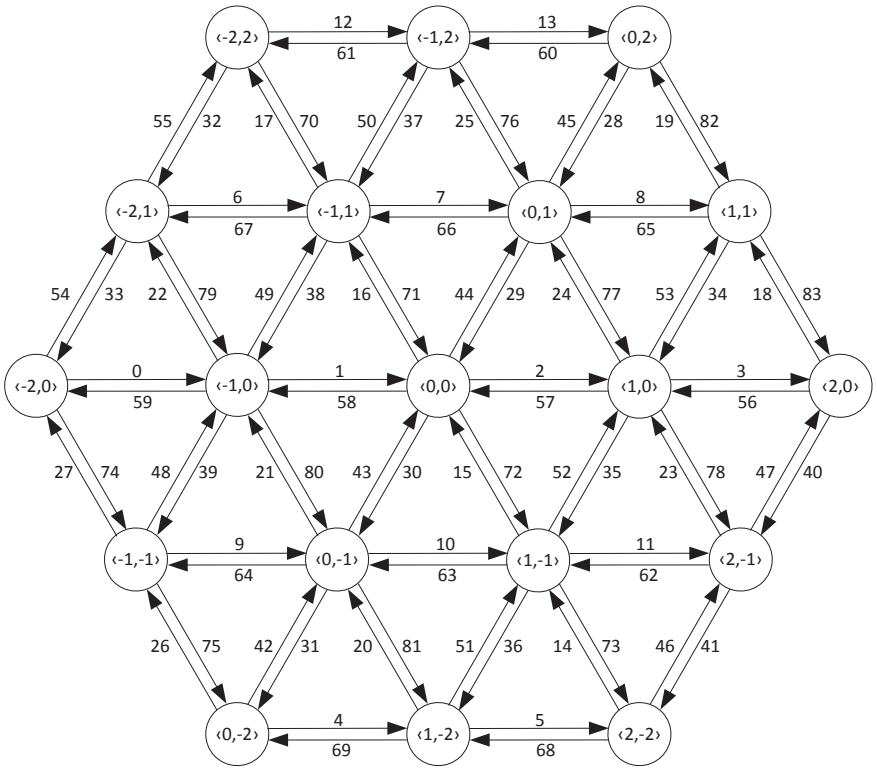


Figure 2.9: Enumeration of an $\mathcal{H}_3$ in deterministic routing.

2.4.2 Routing in Hexagonal Torus Networks

In the last subsection we give a deterministic routing algorithm for hexagonal meshes. In this subsection, we use additional virtual channels in order to apply that algorithm for hexagonal tori.

The routing algorithm for hexagonal networks proposed in this section is based on the concept of dateline classes introduced by Dally and Towles in [17] combined with the shortest path algorithm for hexagonal networks in [25].

The virtual channel dependence graph in Fig. 2.10 depicts how a cycle in a four-node ring can be broken by using two virtual channels $c0$ and $c1$ per physical channel. Each packet begins by taking the $c0$ channels and then switches to the $c1$ channel if the dateline is crossed. We apply this idea to hexagonal torus networks

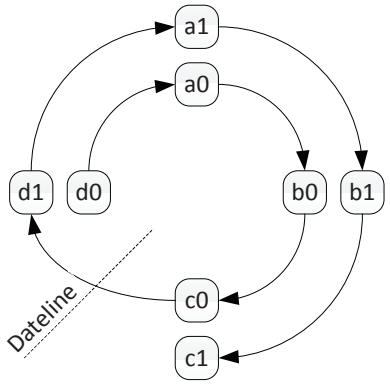


Figure 2.10: Virtual channel dependence graph in a four node ring using the concept of dateline classes. (This figure is taken from [17].)

by adding extra virtual channels per each physical channel and considering each wraparound channel as a dateline.

2.4.3 A Deterministic Algorithm in Hexagonal Torus Networks

Consider the deterministic algorithm defined by the turns in Fig. 2.8. The following Lemma shows that any message using the deterministic algorithm will use at most two wraparound channels.

Lemma 2.2. *Any message routed by either the deterministic algorithm or by the partially adaptive routing algorithm in hexagonal tori will use at most two wraparound channels.*

Proof. Once a message uses a wraparound channel in one direction, it cannot use any more wraparound channels in that direction. Consider the hexagonal network $\mathcal{H}_n$ generated by $\alpha = n + (n-1)\omega$. The diameter of this network is $n-1$ and the minimum number of nodes along a given direction is n. (For example, the minimum number of nodes along a row is n and this occurs at the bottom and the top rows.) Thus if a message uses more than one wraparound channel in a direction to reach its destination this path length will be greater than n and so it

is not a shortest path. Since message-types consist of at most two directions, they can use at most two wraparound channels, one in each direction. □

Two classes of virtual channels (VC0 and VC1) are used in this algorithm Regular messages utilize channels in VC0. Wraparound messages which are the messages that use one or more wraparound channels use channels in VC1 when they are injected to the network. As soon as they cross all the required wraparound channels, they use channels in VC0.

Theorem 2.3. *The deterministic routing algorithm defined by the turns in Fig. 2.8 is deadlock-free in hexagonal tori.*

Proof. We assign numbers to the virtual channels in a way that all the numbers for VC1 channels are less than those of VC0 channels. In this way, the cross dependencies between the two classes don't cause deadlock because the way the channels are assigned guarantees that no cycle will occur between the two classes. Thus, it suffices to prove that there exists no cyclic channel dependency on each one the classes. To number the VC0 channels we use the same method as described in subsection 2.4.1. The channels in VC1 are numbered in the increasing order determined by the Hamiltonian cycle in each direction. The numbering at each direction starts from the origin and ends at the origin along the corresponding Hamiltonian cycle. Since we are considering hexagonal tori, we also assign numbers to the wraparound channels as they are visited along the corresponding Hamiltonian cycle. The numbering starts along the Hamiltonian cycle in the ω^0 direction and continues along the Hamiltonian cycles in the remaining directions with the following order: ω^2, ω^4, ω^1, ω^3, ω^5.

In this way, it is guaranteed that the numbers are increasing along each direction for the VC1 channels except for the "rows" that contain the origin. (Here "row" can be in any of the six directions). In such rows the number assigned to the channels in the first half of the row is greater than the number assigned to the channels in the second half of the row. (The direction along the row specifies the two halves.) However, no message originates from the first half of the row uses

a wraparound channel because the path length of such message is greater than n in $\mathcal{H}_n$. By the shortest path algorithm any message length has to be less than or equal to $n-1$.

□

2.5 Partially Adaptive Routing in Hexagonal Mesh and Torus Networks

In this section we propose partially adaptive routing algorithms for hexagonal meshes and tori.

2.5.1 A Partially Adaptive Algorithm in Hexagonal Mesh Networks

Although the deterministic algorithm proposed in Section 2.4.1 is deadlock-free, it offers no adaptivity. Prohibiting only one turn from each of the two cycles in Figure 2.7 yields a more adaptive mechanism (which is still not fully adaptive.) In general, adaptive routing algorithms are more desirable than deterministic routing algorithms as adaptivity provides more path diversity. However, deterministic routing algorithms are easy to implement [17, Chapter 8.2].

There are 36 possible ways to eliminate two turns from each one of the two hexagonal cycles, but not all of them yields a deadlock-free routing algorithm. For example, consider allowing the five turns shown in Figure 2.11(a) with the five turns in one of Figure 2.11(b) through Figure 2.11(g). When the turns in Figure 2.11(g) are allowed no type-6 message is routed in the network and the cycle in Figure 2.12 can occur in the associated resource dependence graph.

In addition, whenever all message-routings of four consecutive message-types are allowed cycles can be generated in the resource dependence graph. (This happens in $\mathcal{H}_n$ for all $n \geq 4$). For instance, consider the partially adaptive routing algorithm that uses only the turns in Figure 2.11(a) and Figure 2.11(b). This is the same as Figure 2.11(f) and Figure 2.11(g) where here type-1 and type-6 messages are deterministic and the remaining message-types can be routed fully

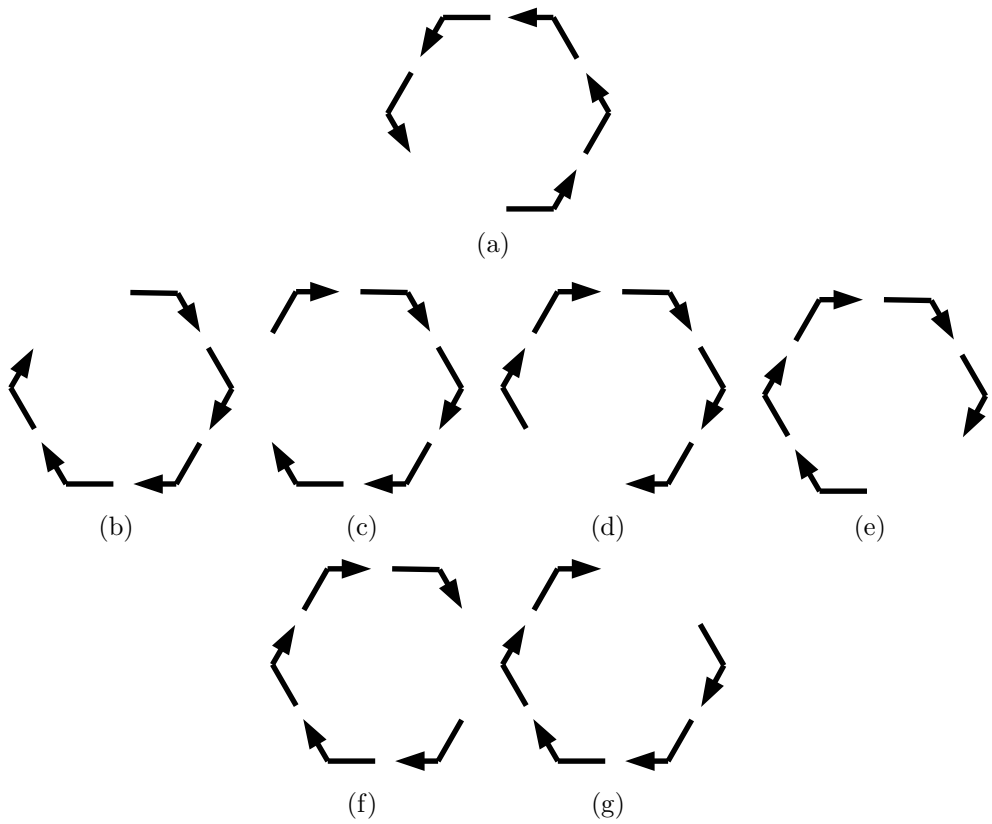


Figure 2.11: The revised turn model for an hexagonal network.

adaptively. Figure 2.13 shows an example of a cycle in the resource dependence graph. Similarly, cycles exists in the resource dependence graph of the routing algorithm that uses the turns in Figure 2.11(a) and Figure 2.11(f).

We consider the partially adaptive routing algorithm that uses only the turns in Figure 2.11(a) and Figure 2.11(d). Two turns are not allowed in this algorithm; that is, the routing is deterministic for messages of types 3 and 6. In order to prove this routing is deadlock-free, we will impose a total ordering on the network channels so that every packet is routed along channels in a strictly increasing (or decreasing) order. This guarantees that the partially adaptive algorithm is deadlock-free [15].

A sample enumeration for $\mathcal{H}_3$ (using the ten turns in Figure 2.11(a) and Fig-

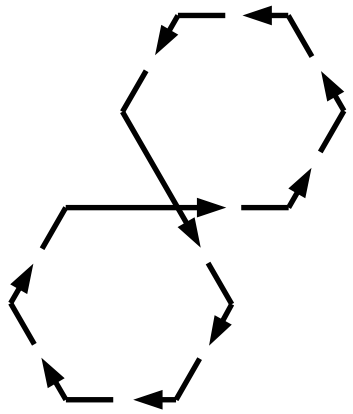


Figure 2.12: Combining Figure 2.11(a) and 2.11(g) causes deadlock.

ure 2.11(d)) is given in Figure 2.14. The choice of allowed turns gives us a clue that all ω^2, ω^1 and ω^0 channels need to be numbered first. (Since we are numbering from bottom to top the ω^1 channels of a given row must be numbered before the ω^2 and ω^0 channels of the next row so that the values along the turns from ω^1 to ω^2 as well as from ω^1 to ω^0 increase along the corresponding turns.) In addition, the values of the channels in each direction must increase along that direction. Starting from the node $\langle 0, -2\rangle$ in Figure 2.14, we number its $\omega^2, \omega^1, \omega^0$ channels as 0, 1, 2, respectively. Moving to the next node in the ω^2 direction (the node $\langle -1, -1\rangle$) its $\omega^2, \omega^1, \omega^0$ channels are assigned 3, 4, 5, respectively. For the next node $\langle -2, 0\rangle$ in the ω^2 direction, the ω^2 channel of $\langle -2, 0\rangle$ is a wraparound channel and this is not given a number; assign 6,7 to its ω^1, ω^0 channels.

The node $\langle 1, -2\rangle$ is on the East side of the node $\langle 0, -2\rangle$, and its ω^2, ω^1 and ω^0 channels are assigned 8, 9, and 10 respectively. This continues through the next NE nodes, each time assigning to the ω^2, ω^1 and ω^0 channels the next set of integers. Continuing this process by taking next set of the nodes along the ω^2 direction, the node $\langle 1, 1\rangle$ is the last one to be considered and 41 is the highest number given to a channel. Proceeding to the node $\langle 0, 2\rangle$ we assign 42, 43, and 44 to the channels ω^5, ω^4 and ω^3 respectively. From here, the ω^5, ω^4 and ω^3 channels (as available) of the nodes $\langle 1, 1\rangle$ and $\langle 2, 0\rangle$ (in that order) along the ω^5 direction

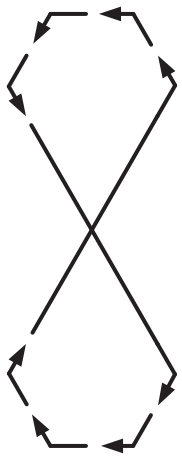


Figure 2.13: Combining Figure 2.11(a) and 2.11(b) causes deadlock.

are assigned the next set of integers, namely, 45, 46, 47, 48 and 49. This process is continued by taking the next set of nodes along the ω^5 direction, going from top to bottom. Figure 2.14 shows the numbers assigned to the channels of $\mathcal{H}_3$. Using only the turns allowed in Figure 2.11(a) and Figure 2.11(d), in Theorem 2.4 it is proved that a packet can be routed from any pair of source and destination using a strictly increasing sequence of channel numbers.

Let the total number of channels in each direction be c and thus the total number of channels is $6c$. Using this fact, the channels in $\mathcal{H}_n$ are numbered in a two-phase process as follows:

In the first phase, all the ω^2, ω^1 and ω^0 channels are given distinct numbers from 0 to $3c-1$, according to the following ordering. Visit the nodes along the ω^2 direction starting at the node $\langle 0, -(n-1)\rangle$. For each node the ω^2, ω^1 and ω^0 channels (if they exist) are assigned the next set of consecutive integers. Continue the process each time starting at the nodes $\langle 1, -(n-1)\rangle, \ldots, \langle n-1, -(n-1)\rangle$, $\langle n-1, -(n-1)+1\rangle, \ldots, \langle n-1, 0\rangle$.

In the second phase, all the ω^5, ω^4 and ω^3 channels are assigned distinct numbers from $3c$ to $6c-1$ as follows. Order the nodes along the ω^5 direction starting at the node $\langle 0, n-1\rangle$. For each node, the ω^5, ω^4 and ω^3 channels (if they exist) are assigned

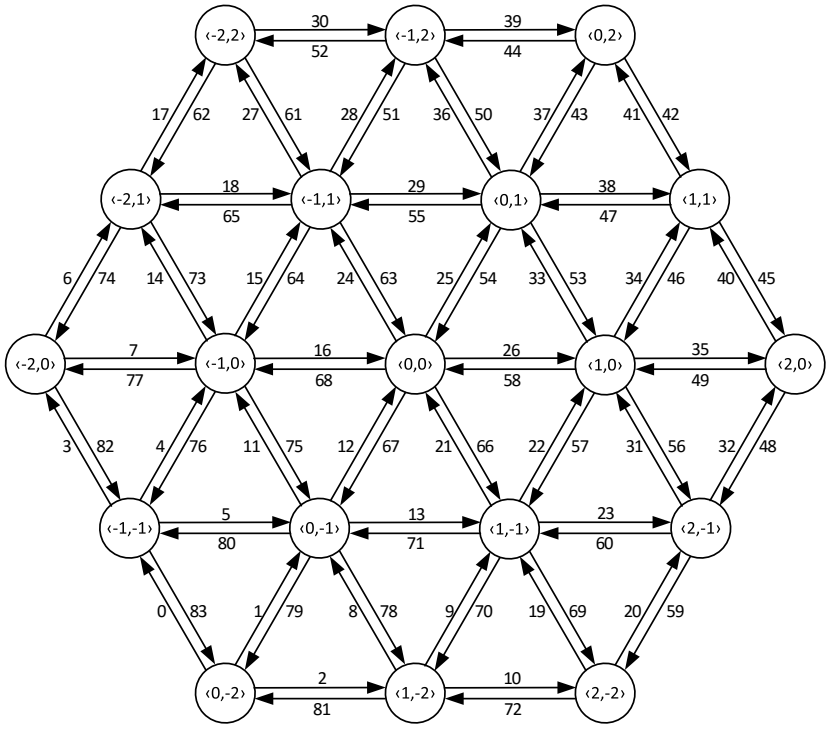


Figure 2.14: Our numeration of the edges of $\mathcal{H}_3$ in partially adaptive routing.

the next set of consecutive integers. Continue the process each time starting at the nodes $\langle -1, n-1\rangle, \ldots, \langle -(n-1), n-1\rangle, \langle -(n-1), n-2\rangle, \ldots, \langle -(n-1), 0\rangle$.

Theorem 2.4. *If only the turns in Figure 2.11(a) and Figure 2.11(d) are allowed the resulting algorithm is deadlock-free in hexagonal meshes.*

Proof. Each interior node $\langle i, j\rangle$ has at most six outgoing channels and at most six incoming channels connecting the node to its neighbors. An example with six neighbors is given in Figure 2.15. The channels are identified by the number of their source node and their direction. For example, the direction 0 channel from the node $\langle i, j\rangle$ to the node $\langle i+1, j\rangle$ is c_{0ij}. A turn is identified from a channel to another channel by an arrow. For instance, $c_{0ij} \rightarrow c_{1(i+1)j}$ shows the turn from the channel 0 of the node $\langle i, j\rangle$ to the channel 1 of the node $\langle i+1, j\rangle$.

As we stated in the numbering system, the values of channels in each direction

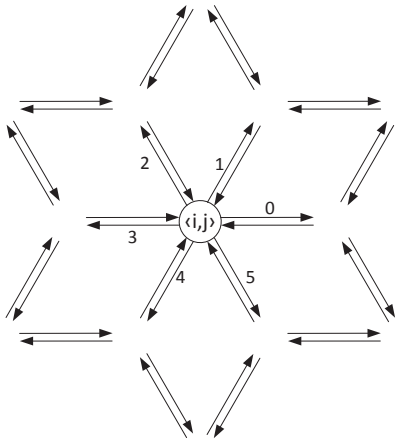


Figure 2.15: Deadlock-freedom proof.

increase along that direction. Thus, we only need to prove that the values of channels in an allowed turn increase along that turn. Ignoring the two prohibited turns, there are 12 turns originating from the node $\langle i, j \rangle$ as follows:

$1 : c_{0ij} \rightarrow c_{5(i+1)j}$ $\quad$ $2 : c_{0ij} \rightarrow c_{1(i+1)j}$

$3 : c_{1ij} \rightarrow c_{0i(j+1)}$ $\quad$ $4 : c_{1ij} \rightarrow c_{2i(j+1)}$

$5 : c_{2ij} \rightarrow c_{1(i-1)(j+1)}$ $\quad$ $6 : c_{2ij} \rightarrow c_{3(i-1)(j+1)}$

$7 : c_{3ij} \rightarrow c_{2(i-1)j}$ $\quad$ $8 : c_{3ij} \rightarrow c_{4(i-1)j}$

$9 : c_{4ij} \rightarrow c_{3i(j-1)}$ $\quad$ $10 : c_{4ij} \rightarrow c_{5i(j-1)}$

$11 : c_{5ij} \rightarrow c_{4(i+1)(j-1)}$ $\quad$ $12 : c_{5ij} \rightarrow c_{0(i+1)(j-1)}$

Note that this is a generic node since not every node has all six outgoing channels and some turns cannot occur.

From the above turns, turns numbered 7 and 12 are not allowed by the current combinations of the turns. Since we enumerate all channels along ω^0, ω^1 and ω^2 directions before the rest of the channels the turns 1 and 6 preserve the increasing order of channel values. In the first phase of the enumeration process the channels along the ω^0, ω^1 and ω^2 directions for the node $\langle i, j \rangle$ are enumerated before the channels of the nodes $\langle i + 1, j \rangle$, $\langle i, j + 1 \rangle$, and $\langle i - 1, j + 1 \rangle$, guaranteeing the increasing order of channel values in the turns 2, 3, 4, and 5. Similarly, the channel values in the remaining turns (8, 9, 10 and 11) are numbered in an increasing order.

This is because in the second phase of the enumeration process the channels along ω^3, ω^4 and ω^5 directions for the node $\langle i, j \rangle$ is enumerated before the corresponding channels of the nodes $\langle i-1, j \rangle$, $\langle i, j-1 \rangle$, and $\langle i+1, j-1 \rangle$. Thus, the partially adaptive routing algorithm is deadlock-free. □

2.5.2 A Partially Adaptive Algorithm in Hexagonal Torus Networks

In this subsection we extend the algorithm in Section 2.5.1 to hexagonal tori. This means that only the turns in Figure 2.11(a) and Figure 2.11(d) are allowed by the algorithm in this subsection.

By Lemma 2.2 no message can take more than two wraparound channels in hexagonal tori. We use three classes of virtual channels (VC0, VC1, and VC2). Regular messages utilize VC0. Wraparound messages use VC2 when they are injected to the network. As soon as they cross a wraparound edge, they use VC1 if one more wraparound edge is required for them to reach their destination. Wraparound messages switch to VC0 when no more wraparound edge is required to reach their destination.

Theorem 2.5. *If only the turns in Figure 2.11(a) and Figure 2.11(d) are allowed the resulting algorithm is deadlock-free in hexagonal tori.*

Proof. VC2 channels and VC1 channels are numbered in a way similar to that described in subsection 2.5.1 with one modification. The process is as follows. Consider $\mathcal{H}_n$. Let $N = 3n^2 - 3n + 1$ be the total number of nodes. Since each node has six outgoing channels (and six incoming channels) the total number of channels is $6N$. That is, there are N channels (including the wraparound channels) in each direction. Similar to the subsection 2.5.1, the channels in $\mathcal{H}_n$ are numbered in a two-phase process as follows.

In the first phase, all the ω^2, ω^1 and ω^0 channels are given distinct numbers from 0 to $3N - 1$, according to the exact same ordering as given in the first phase described in subsection 2.5.1. Note that here each node has three channels in each of the three directions. Some of the channels might be wraparound channels.

In the second phase, all the ω^5, ω^4 and ω^3 channels are assigned distinct numbers from $3N$ to $6N-1$ according to the exact same ordering given in the second phase described in subsection 2.5.1. Again, every node has three channels in each of the three directions. Some of the channels might be wraparound channels.

After numbering the VC2 channels from 0 to $6N-1$ as described above, we number the VC1 channels from $6N$ to $12N-1$ in the same way. Finally, we number the VC0 channels from $12N$ to $12N+6c-1$. (Remember that c is the total number of channels in one direction in a hexagonal mesh network.) All VC0 channels are numbered exactly as described in subsection 2.5.1.

Note that, when we were proving the Theorem 2.4 we mentioned that not every node has all six outgoing channels. But even if every node has all six outgoing channels the proof remains the same. Thus, based on Theorem 2.4 there is a total ordering within each class of virtual channels. If a message wants to switch its class of virtual channel, then it always goes from a higher numbered class to a lowered numbered class based on the proposed algorithm. Therefore, there is a total ordering considering all three classes of virtual channels which means that no cyclic dependencies will be created.

□

2.6 Conclusion

This chapter proposes two new deadlock-free and minimal routing algorithms for hexagonal mesh and torus interconnection networks. By prohibiting certain turns in the resource dependence graph, the proposed algorithms are deadlock-free. In addition, since they are minimal routings, no livelock can happen in the network. Some additional virtual channels per physical channel are used to remove inherent cycles caused by the wraparound edges.

Although the partially adaptive routing algorithms proposed in this chapter offer good adaptivity, they are not still fully adaptive. Obtaining fully adaptive routing algorithm for hexagonal mesh and torus networks would be one direction for future work. Developing fault-tolerant routing algorithms (that is routing al-

gorithms that can route a packet in existence of faulty nodes or channels) would be another direction for the future.

Chapter 3: Higher Dimensional Gaussian Networks

3.1 Introduction

The original Gaussian networks were proposed [36] as compelling alternatives to classical 2D-torus networks. An on-chip implementation of Gaussian networks was developed in [49]. One advantage of these networks over 2D-torus networks is that they are degree four networks that have a smaller diameter for the same number of nodes. This results in a lower average message latency. One of the reasons for considering higher dimensional toroidal networks is to reduce the diameter-to-nodes ratio further.

In this chapter we develop higher dimensional Gaussian networks [1] . The main advantages of these networks are as follows.

- They are Cayley graphs and so they are node symmetric.
- The average distance of these networks is always less than that of the toroidal networks with the same degree and number of nodes.
- The diameter of these networks is $\sqrt{2}$ times less than that of the toroidal networks with same degree and same number of nodes.
- Because of the above two properties, shortest-path routing algorithms in Gaussian networks always outperform such algorithms in toroidal networks.
- The collection of Gaussian networks is a more general class of networks which includes toroidal networks as a special case.
- It is possible to embed two edge-disjoint k-ary n-cube in an n-dimensional Gaussian network. This means any algorithm developed for a torus network

[1] This chapter has been published in [8, 42]

can be easily transported to the Gaussian network. Furthermore, the network can be decomposed into the maximum number of edge-disjoint Hamiltonian cycles.

In [39] a different type of 2-dimensional Gaussian network, called *hierarchical Gaussian networks*, is constructed which is not node-symmetric.

The rest of the chapter is organized as follows. Section 3.2 briefly reviews important properties of the original Gaussian networks that are relevant to the rest of the paper. The definition and some basic topological properties of higher dimensional Gaussian networks are given in Section 3.3. Next, Section 3.4 provides a shortest-path routing algorithm for the networks, while the distance distribution and the average distance are given in Section 3.5. Some simulation studies comparing the routing algorithms of Gaussian networks and torus networks are given in Section 3.6. Section 3.7 shows how broadcasting can be accomplished. Embedding edge-disjoint toroidal networks, including edge-disjoint Hamiltonian cycles, is described in Section 3.8. Finally, the summarizing conclusions are given in Section 3.10.

3.2 A Review of One-dimensional Gaussian Networks

Gaussian networks are wraparound networks of degree four in which the generating rectangle for the classical torus network is replaced by any square whose vertices have integer coordinates. We may assume the southeastern vertex of the square is the origin and its northeastern vertex is (a, b) where a, b are non-negative integers. This square can be used to tile the plane, as illustrated in Figure 3.1 for $(a, b) = (3, 5)$. The way the Gaussian network is presented in Figure 3.1 is called *half-open square* representation [24].

We now describe the Gaussian network generated by this square. We will denote the network by G_{a+bi} where the complex number $a + bi$ (for $i^2 = -1$) is used to represent the point (a, b). The nodes of the network are the origin and all integer points lying either within the square or on the interior of one of the

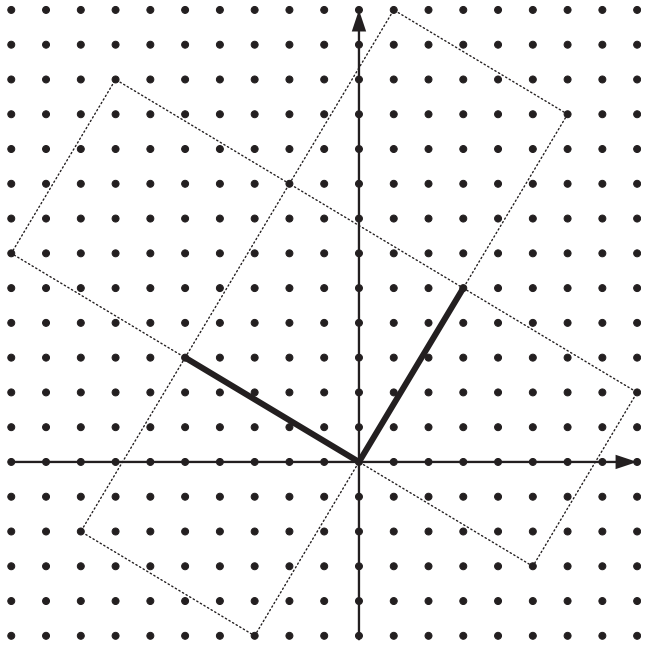


Figure 3.1: The basic square and surrounding tiles for $(a, b) = (3, 5)$.

two sides containing the origin. There are $a^2 + b^2$ nodes in G_{a+bi}. As with torus networks, Gaussian networks have two types of links: regular and wraparound. Regular links connect neighboring points within the basic square. For instance, in Figure 3.1, the node $(0, 2)$ has four regular neighbors whereas $(2, 5)$ has only two regular neighbors. The wraparound links are obtained by pasting each of the two sides containing the origin to its opposite side, forming a torus with a rotated square grid. This is illustrated in Figure 3.2 for G_{3+5i}.

An easy way to find the wraparound edges follows from the fact that the basic square tiles the plane with no overlapping. For instance, to find the two wraparound edges at $(2, 5)$ first note that as a point in the plane $(2, 5)$ is adjacent to the points $(1, 5)$, $(2, 4)$, $(3, 5)$, $(2, 6)$. Only the first two points are nodes in G_{3+5i}. Referring to the tiling in Figure 3.1, the other two points can be translated to the points $(0, 0)$ and $(-1, 1)$ in the same position in the basic square. These are the wraparound neighbors of $(2, 5)$.

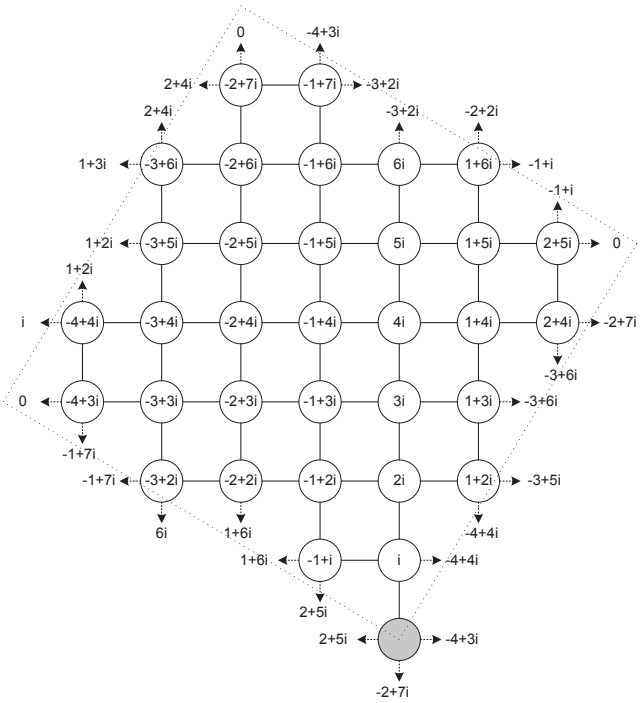


Figure 3.2: A picture of G_{3+5i}.

This geometric structure can be rephrased in algebraic terms based on the representation of an integer point (x, y) in the plane as the complex number $x+yi$. The set of such numbers is a subring of the complex numbers, called the ring of Gaussian integers. In the tiling for $\alpha = a + bi$ the tile obtained by translating the basic tile by $x\alpha$ and then by $y\alpha i$ has SE corner $(x+iy)\alpha$. Consequently, the set of all tile corners is the set of all Gaussian multiples of $a+bi$ where the multiple gives the location of the tile. Because of this, the action of translating any node back to the basic square amounts to finding an equivalent representative modulo α. For example, when $\alpha = 3 + 5i$ (refer to Figure 3.1) the integer point $(-4, 10)$ is in the tile obtained by translating using $\alpha+\alpha i = -2+8i$, and the equivalent point in the basic tile is $(-4, 10) - (-2, 8) = (-2, 2)$. From this we see that the nodes of the Gaussian network for α are the representatives of the Gaussian integers modulo α, and $p+qi \pm 1 \pmod{\alpha}$ and $p+qi \pm i \pmod{\alpha}$ are the neighbors of the node $p+qi$.

This argument proves it is a Cayley graph with four generators, and so is a regular node-symmetric network of degree 4. For the remainder of this paper we will refer to G_α as the one-dimensional Gaussian network generated by α. Also, $\mathbb{Z}_\alpha$ will denote the ring of Gaussian integers modulo α, whose size is $\mathcal{N}(a+bi) = a^2 + b^2$ where $\mathcal{N}(\alpha)$ is called the norm of α.

Gaussian networks generated by $\alpha = k+(k+1)i$ (for positive k) are called dense because they contain the maximum number of nodes for a given diameter [38]. The nodes of dense networks are usually arranged according to their distance from 0 rather than using the basic square representation. Figure 3.3 gives this representation for $\alpha = 3 + 4i$.

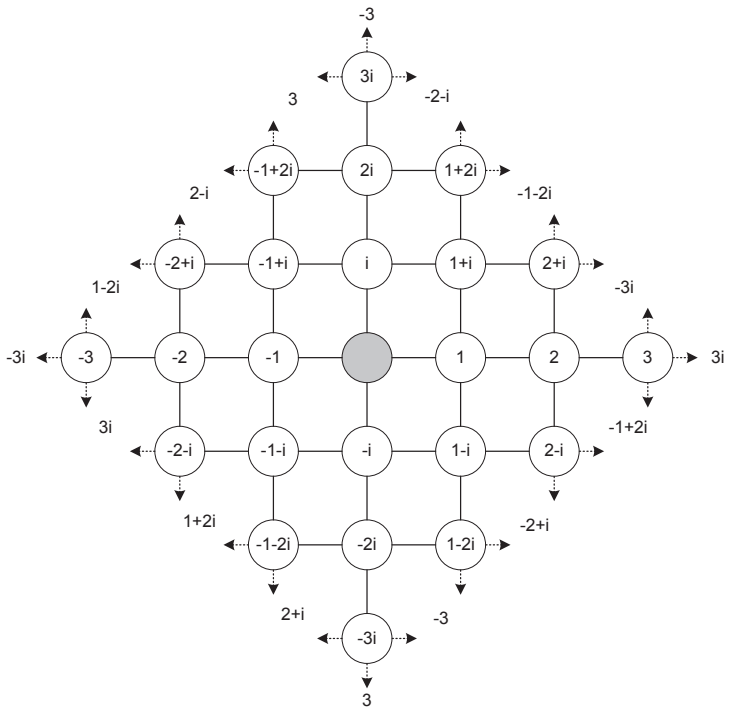


Figure 3.3: The dense Gaussian network G_{3+4i} .

3.3 Higher Dimensional Gaussian Networks

In the case of torus networks, in particular the k-ary n-cube, each node is addressed using n-digit vectors with coordinates in $\mathbb{Z}_k$. Two nodes in this network are adjacent if and only if the node addresses differ in exactly one component by ± 1 mod k. A similar definition can be used to describe an n-dimensional Gaussian network as below.

Definition 3.1. *Let $\alpha = a + bi \in \mathbb{Z}_\alpha$. The nodes in an n-dimensional Gaussian network, denoted by $G_\alpha^{(n)}$, are represented by n-digit vectors whose components are Gaussian integers modulo α. Two nodes are adjacent if and only if their node addresses differ in exactly one position by $\pm 1 \mod \alpha$ or $\pm i \mod \alpha$.*

The theory remains the same even if the components of the nodes are allowed to come from different α's.

One helpful observation is that $G_\alpha^{(n)}$ is a Cayley graph since it is generated by the $4n$ basic vectors that have exactly one nonzero component which is equal to either ± 1 or $\pm i$. Thus, $G_\alpha^{(n)}$ is a node-symmetric network.

Another useful way of describing an n-dimensional Gaussian network is using the concept of the cross product of graphs, which is sometimes called the Cartesian product or simply the product of graphs. It has long been known [5, 20, 33, 48] that the graph-theoretic concept of cross product can be used to obtain higher dimensional interconnection networks. Given two networks N_0, N_1, their cross product $N_1 \otimes N_0$ is the network whose node set is the Cartesian product of the two node sets, and the nodes $(u_1, u_0), (v_1, v_0)$ are adjacent in $N_1 \otimes N_0$ when either $u_1 = v_1$ and u_0 is adjacent to v_0 in N_0 or $u_0 = v_0$ and u_1 is adjacent to v_1 in N_1. Intuitively, the cross product $N_1 \otimes N_0$ has a copy of N_0 at every vertex of N_1 and also a copy of N_1 at every vertex of N_0. For instance, if C_k is a cycle of size k then $C_k \otimes C_k$ is a torus, and $C_{k_1} \otimes C_{k_2}$ is a mixed radix torus.

Since $N_1 \times N_0$ is the set of nodes, the number of nodes in $N_1 \otimes N_0$ is the product of the number of nodes in the factors, and from the adjacency relation the degree of the product is the sum of the degrees of the factors. As has already been

observed in [20, 48], a shortest path can be accomplished in two legs by traversing a shortest path in each component. Therefore, the distances add and the diameter of the cross product is the sum of the diameters of the factors.

Example 3.2. *$G^{(2)}_{2+3i}$ has 169 nodes and is a regular graph of degree 8. Figure 3.4 pictures all links incident to the nodes $(0,0)$ and $(-2,-1+i)$ in $G^{(2)}_{2+3i}$. The node $(0,0)$ is adjacent to 8 nodes, that can be grouped into two sets: $(1,0),(-1,0),(i,0),(-i,0)$ and $(0,1),(0,-1),(0,i),(0,-i)$. Similarly, the node $(-2,-1+i)$ is adjacent to the 8 nodes $(-1,-1+i),(-2i,-1+i),(1-i,-1+i),(2i,-1+i)$ and $(-2,i),(-2,1-i),(-2,2),(-2,-1)$.*

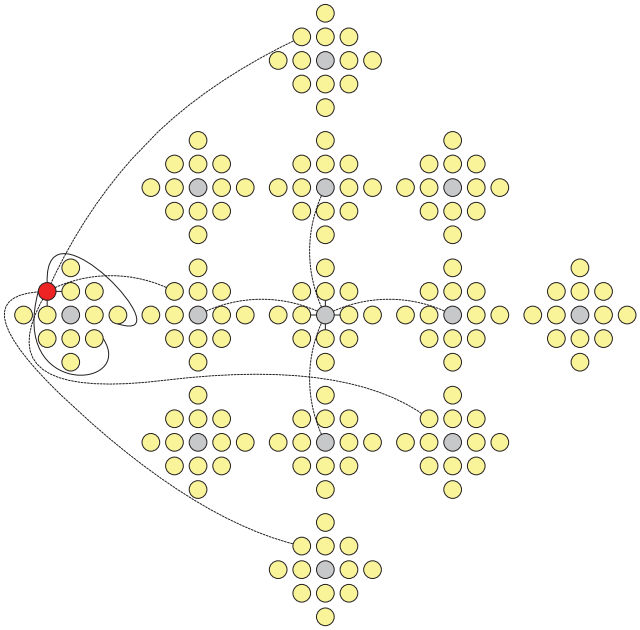


Figure 3.4: All nodes in $G^{(2)}_{2+3i}$ are given, along with all links incident to either $(0,0)$ or $(-2,-1+i)$.

The cross product can be inductively defined for any set of finitely many networks $N_0,\ldots,N_n$ as

$$N_n \otimes N_{n-1} \otimes \ldots \otimes N_0 = N_n \otimes (N_{n-1} \otimes \ldots \otimes N_0) \ ,$$

and this is used to define an n-dimensional Gaussian network $G_\alpha^{(n)}$ as the n-fold cross product of the Gaussian network G_α;

$$G_\alpha^{(n)} = \overbrace{G_\alpha \otimes \ldots \otimes G_\alpha}^{n \text{ times}} = G_\alpha \ \otimes \ \overbrace{(G_\alpha \otimes \ldots \otimes G_\alpha)}^{n-1 \text{ times}}.$$

As before, the theory remains the same even if the components of the nodes are integers modulo different α_j's. Note that for $\alpha = a$ or $\alpha = bi$, $G_\alpha^{(n)}$ is exactly the a-ary n-cube or the b-ary n-cube.

The network $G_\alpha^{(n)}$ has $\mathcal{N}(\alpha)^n = (a^2+b^2)^n$ nodes, since it is the set of all n-tuples with components in G_α. One way to picture $G_\alpha^{(n)}$ is that a copy of $G_\alpha^{(n-1)}$ has been placed on each node of G_α.

Our earlier discussion also shows that the degree of each node is $4n$ and that a shortest path in $G_\alpha^{(n)}$ can be constructed by taking any n-leg combination of shortest paths in each component. Consequently, the distance is the sum of the distances in the components, and the diameter of $G_\alpha^{(n)}$ is n times the diameter of G_α.

3.3.1 Implementation

Gaussian networks like toroidal networks are wraparound topologies in which the size of wraparound edges grows by the network size whereas the regular edges (non-wraparound edges) are supposed to have unitary length. Tori are known to have a folded shape in which the regular edges are doubled in size and the length of the wraparound edges are reduced to be the same as the length of the regular edges in the folded layout [16].

As an attempt to alleviate the problem of long wraparound edges, an elegant folded layout for dense Gaussian networks was presented in [38, 45]. Initially, the nodes of a dense Gaussian network generated by $\alpha = k + (k+1)i$ are arranged in $2k+1$ rows and $k+1$ columns. The first row only contains one node which is chosen to be the origin by the network symmetry. This is illustrated in Figure 3.6(a) for the

dense Gaussian network presented in Figure 3.3 where $k = 3$. In the initial layout, vertical edges correspond to imaginaries and diagonal edges represent reals. Other edges are obtained followed by the adjacency patterns of the Gaussian network.

Then a set of shuffle transformations are performed on the rows according to the following procedure: (The shuffles are given in Figure 3.5)

- For rows $1 \leq i \leq k+1$ rotate (clockwise) $\lfloor \frac{i-1}{2} \rfloor$ nodes and then perform shuffle A to odd rows and shuffle B to even rows.
- For rows $k+2 \leq i \leq 2k+1$ rotate (clockwise) $\lfloor \frac{i}{2} \rfloor$ nodes and then perform shuffle B to odd rows and shuffle A to even rows.

Shuffle A:

$$x' = \begin{cases} 2x-1 & \text{if } x \leq (n+1)/2 \\ 2n-2x+2 & \text{if } x > (n+1)/2 \end{cases}$$

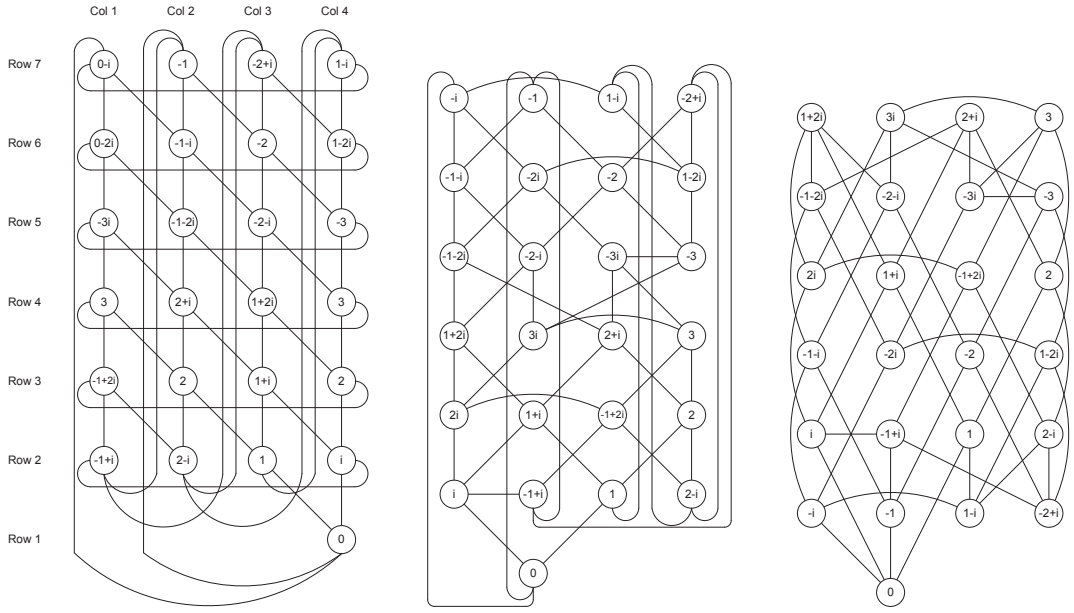
Shuffle B:

$$x' = \begin{cases} 2x & \text{if } x < (n+1)/2 \\ 2n-2x+1 & \text{if } x \geq (n+1)/2 \end{cases}$$

Figure 3.5: The shuffles transform a node in position x to a node in position x' in one of the n positions of the corresponding row or column.

The layout after row transformations is shown in Figure 3.6(b) for G_{3+4i}. Finally all columns are shuffled according to shuffle A. Figure 3.6(c) shows the final layout for G_{3+4i}. In the final layout, no edge has a wire length greater than $\sqrt{5}$. Although in a folded torus all edges are doubled in length but the approximate diameter of a folded Gaussian network is $L\sqrt{\frac{5N}{2}}$ whereas that of a folded torus is $L\sqrt{4N}$, where N is the number of nodes and L is the unit of length. In terms of approximate average distance a folded torus is barely superior. For more information refer to [38, 45].

We use the exact set of transformations in order to reduce the length of the long wraparound edges in higher dimensional Gaussian networks. For example, the folded layout of $G_\alpha^{(2)}$ is obtained in two steps. Recall that each super node in



(a) Initial layout

(b) layout after row tranformations

(c) Final layout

Figure 3.6: Folded dense Gaussian network for the Gaussian network generated by $\alpha = 3 + 4i$.

$G_\alpha^{(2)}$ is a G_α. First, the shuffle transformations explained above are used to fold all the Gaussian networks G_α. Then, the shuffle transformations transform all the super nodes in their final place.

Similarly, $G_\alpha^{(n)}$ can be transformed to a folded layout in n steps to reduce the wire length.

3.4 Shortest-path Routing in Higher Dimensional Gaussian Networks

As noted in the last section, a shortest path between $A = (a_{n-1}, \ldots, a_0)$ and $B = (b_{n-1}, \ldots, b_0)$ in $G_\alpha^{(n)}$ can be obtained by using a shortest path routing from a_i to b_i in each factor of the cross product. In this section we propose an optimal routing algorithm using the optimal routing in G_α as given in [24, 36].

Since G_α^n is node-symmetric, to find a shortest path from A to B we can first find one from 0 to $C = B - A$ and then translate back by A, where all operations are done modulo α in each component. In [24] a shortest path from 0 to c_i is found by calculating the minimal Gaussian distance from c_i to the four corners of the basic square. More details can be found in [24, Section VI]. The previous best Gaussian routing algorithm [36, Algorithm 1] required thirteen comparisons.

Before giving our routing algorithm in $G_\alpha^{(n)}$ we first illustrate the algorithm with an example.

Example 3.3. *Let $\alpha = 3 + 4i$. Our algorithm finds a shortest path from $A = (-1 + i, 1 + 2i)$ to $B = (-1 - i, -1 + i)$ in the two-dimensional $G_\alpha^{(2)} = G_\alpha \otimes G_\alpha$. First, $C = B - A = (-2i, -2 - i)$ is calculated using the modulo α difference in each component. A shortest path is then obtained by correcting the offset C, changing one component at a time using the sequence of links specified by the routing algorithm in G_α. Thus, one shortest path from A to B is*

$$\begin{aligned} A = &(-1 + i, 1 + 2i), (-1, 1 + 2i), (-1 - i, 1 + 2i), \\ &(-1 - i, 2i), (-1 - i, -1 + 2i), (-1 - i, -1 + i) = B \ . \end{aligned}$$

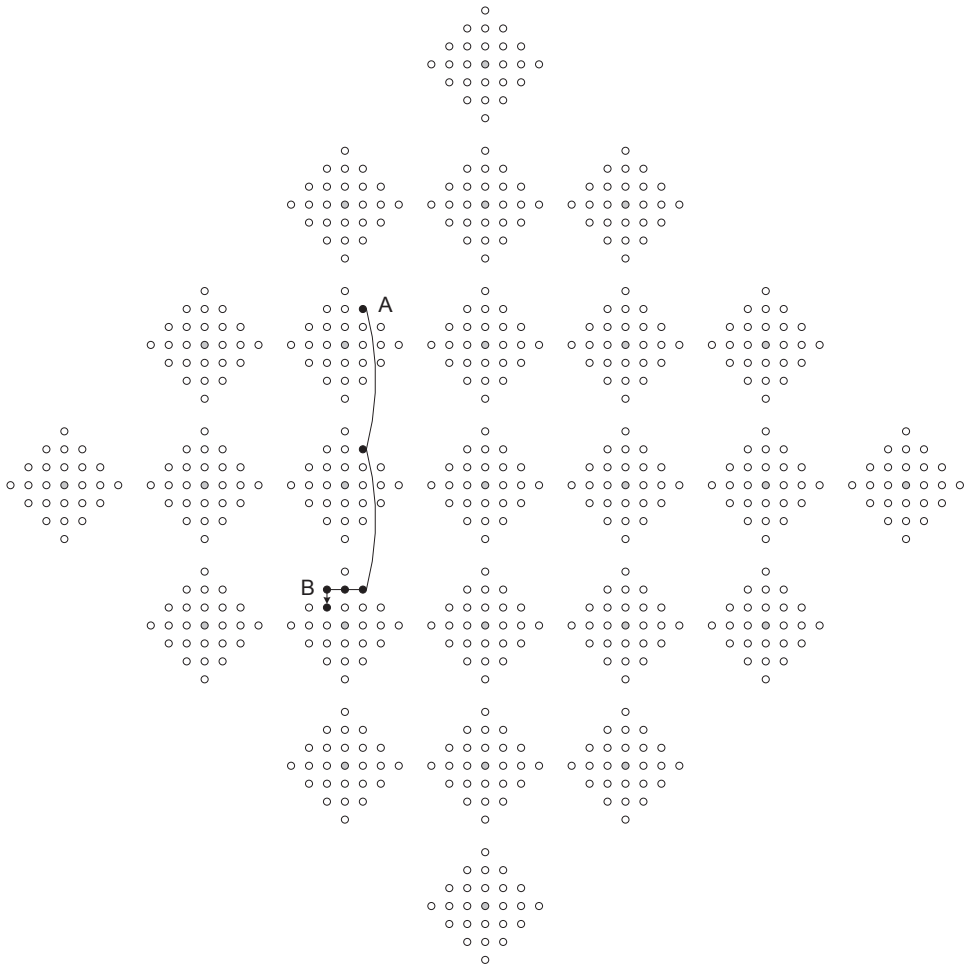


Figure 3.7: The path described in Example 3.3.

The distance from A to B equals $2 + 3 = 5$.

Suppose $v = (v_{n-1}, v_{n-2}, \ldots, v_1, v_0)$ and $w = (w_{n-1}, w_{n-2}, \ldots, w_1, w_0)$ are respectively the source and the destination nodes. Then, the routing algorithm is done by correcting the offset one coordinate at a time in some particular order. The algorithm is described below.

Algorithm 3.4. *Shortest path routing for n-dimensional Gaussian network $G_\alpha^{(n)}$ where $\alpha = a+bi$ $(0 \leq a \leq b$ and $a, b \in \mathbb{Z})$ is the network generator.*

Input: *Two nodes $v = (v_{n-1}, v_{n-2}, \ldots, v_1, v_0)$ and $w = (w_{n-1}, w_{n-2}, \ldots, w_1, w_0)$ where $v_j, w_j \in \mathbb{Z}_\alpha$.*

Output: *A shortest path from v to w.*

1: ***for*** $j = n - 1$ *downto* 0 ***do***
2: *Let* $u_j = ((w_j - v_j) \mod \alpha) = x + yi$.
3: ***while*** $u_j \neq 0 + 0i$ ***do***
4: *Travel x horizontally and y vertically in the $j - th$ coordinate.*
5: ***end while***
6: ***end for***

3.5 The Distance Distribution

Since every n-dimensional Gaussian network is node-symmetric the distance distribution charts the number of nodes at various distances from 0. All nodes at distance t from the origin in $G_\alpha^{(n)}$ can be found by multiplying the nodes at distance one in G_α times the nodes at distance $t-1$ in $G_\alpha^{(n-1)}$, plus nodes at distance 2 in G_α times nodes at distance $t-2$ in $G_\alpha^{(n-1)}$, and so on. Therefore, if $d^{(n)}(t)$ is the number of nodes at a distance of t from 0 in an n-dimensional Gaussian network $G_\alpha^{(n)}$ then the values can be found recursively (with respect to the dimension n) using the equation

$$d^{(n)}(t) = \sum_{j=0}^{nd} d^{(1)}(j) \times d^{(n-1)}(t-j)$$

since the diameter of $G_\alpha^{(n)}$ equals nd where d is the diameter of G_α.

Example 3.5. *The dense network G_{2+3i} has $\mathcal{N}(2+3i) = 13$ nodes, its diameter equals 2, and $d^{(1)}(1) = 4, d^{(1)}(2) = 8$. The distance distributions of the nodes for the one-, two-, and three-dimensional Gaussian networks are given in Table 3.1. For example, the number of nodes at distance 3 in $G_{2+3i}^{(3)}$ is denoted by $d^{(3)}(3)$ and equals*

$$d^{(1)}(0)d^{(2)}(3) + d^{(1)}(1)d^{(2)}(2) + d^{(1)}(2)d^{(2)}(1) + d^{(1)}(3)d^{(2)}(0)$$

which is

$$1 \times 64 + 4 \times 32 + 8 \times 8 + 0 \times 1 = 256 \ .$$

Table 3.1: Distance distributions for $G_{2+3i}^{(n)}$ when $n = 1, 2, 3$.

$d^{(n)}(t)$	$n=1$	$n=2$	$n=3$
$t=0$	1	1	1
$t=1$	4	8	12
$t=2$	8	32	72
$t=3$	0	64	256
$t=4$	0	64	576
$t=5$	0	0	768
$t=6$	0	0	512
Total number of nodes	13	169	2197

The distance distribution can also be found using a matrix recurrence. We first note that since $d^{(n)}(t) = 0$ for all $t > nd$, we are interested in calculating the components of $\mathbf{D_j} = (d^{(j)}(nd), \ldots, d^{(j)}(0))$ for $j = n$. Then the vector $\mathbf{D_1}$ records the distance distribution for G_α which is given in [36, Theorems 10, 11]. Shortest paths of length t are obtained using any combination of shortest paths from each component for which the lengths add to t. Therefore, each $\mathbf{D_{j+1}} = \mathbf{D_j}A$ where A is the square lower triangular diagonal-constant matrix whose first column is the reverse of $\mathbf{D_1}$, and $\mathbf{D_{j+1}} = \mathbf{D_1}A^j$. For Example 3.5,

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 4 & 1 & 0 & 0 & 0 & 0 & 0 \\ 8 & 4 & 1 & 0 & 0 & 0 & 0 \\ 0 & 8 & 4 & 1 & 0 & 0 & 0 \\ 0 & 0 & 8 & 4 & 1 & 0 & 0 \\ 0 & 0 & 0 & 8 & 4 & 1 & 0 \\ 0 & 0 & 0 & 0 & 8 & 4 & 1 \end{pmatrix}$$

3.5.1 Average Distance

For simplicity, assume the generator is dense; that is, $\alpha = k + (k+1)i$ for some integer $k \geq 1$. A consequence of Theorem 3.2 in [48] is that the average distance in $G_\alpha^{(n)}$ equals n times the average distance in G_α. Here we introduce the following simpler proof of this result.

By node-symmetry, the average distance in the network $G_\alpha^{(n)}$ is obtained by finding the average distance from the node $(0, 0, \ldots, 0)$ to all other nodes. Arranging the $\mathcal{N}(\alpha)^n$ node addresses in any order in an array with the node $(0, 0, \ldots, 0)$ as the first row, the sum of the distances of each node from the node $(0, 0, \ldots, 0)$ is the sum of the distances of each coordinate from 0. Since each column of the array contains every element of $G_\alpha^{(n)}$ exactly $\mathcal{N}(\alpha)^{n-1}$ times, the average n-dimensional distance is n times the average distance in G_α.

It is known [36] that in dense networks generated by $\alpha = k + (k+1)i$ there are exactly $4t$ nodes at a distance of t from the node 0 for every $t = 0, \ldots, k$. Therefore, the average distance in dense G_α equals

$$\frac{4(1^2 + 2^2 + \ldots + k^2)}{\mathcal{N}(\alpha)} = \frac{4k(k+1)(2k+1)}{6 \times (2k(k+1)+1)} < \frac{2k+1}{3}$$

for all $\alpha = k + (k+1)i$. This implies that the average distance in $G_\alpha^{(n)}$ is less than $2nk/3$.

3.5.2 Comparisons with torus networks

We continue to assume the n-dimensional Gaussian network is generated by $\alpha = k + (k+1)i$ with $\mathcal{N}(\alpha)^n = (k^2 + (k+1)^2)^n$ nodes. Since the degree of each node in $G_\alpha^{(n)}$ is $2n$, an $2n$-dimensional torus is used for the comparisons.

For the comparison of average distances we use the $2n$-dimensional torus network with the side in each dimension equal to $(2k(k+1)+1)^{1/2}$. The average distance of this torus network is

$$\frac{2n \times \sqrt{2k(k+1)+1}}{4} > \frac{nk}{\sqrt{2}}$$

for all k. Therefore, the average distance in $G_\alpha^{(n)}$ is $\sqrt{8}/3 \approx 0.942$ less than the average distance in an equivalent torus network. (In [17] it is noted that the average distance of a k-ary n-cube is $nk/4$ for k even and $(k-1)(k+1)n/4k \approx nk/4$ for k odd.)

We recall that the diameter of n-dimensional Gaussian network with $\alpha = k + (k+1)i$ is nk and the number of nodes equals $(2k^2+2k+1)^n > 2^n k^{2n}$. The $2n$-dimensiona torus network with $2^n k^{2n}$ nodes has a diameter of $n\sqrt[2n]{2^n k^{2n}} = \sqrt{2}nk$. Therefore, the diameter of the torus network is $\sqrt{2}$ times larger than the equivalent n-dimensional Gaussian network. To give one example, there is a 2-dimensional Gaussian network of degree 8 with $40,000$ nodes whose diameter is 20, while the diameter of a 40,000-node torus with degree 8 must be at least $2 \times 10\sqrt{2} \approx 28$.

3.6 Simulation Results

This section compares the performance of the routing algorithm given in Section 3.4 for higher dimensional Gaussian networks with a routing algorithm in the higher dimensional torus networks. Each physical channel utilizes two virtual channels: The first virtual channel is used by messages not traversing any wraparound link, whereas the second virtual channel is used by the messages that need to take a wraparound link to be delivered to their destination.

For this comparison, we have performed a simulation in which the average message latency is the performance metric. Our simulation experiments are performed using XMulator [40]. The XMulator is a complete, flit-level, event-based, and extensively detailed package for simulation of interconnection networks which can simulate different interconnection networks with arbitrary topology, switching methods, routing algorithms, and even in the presence of faults.

For the sake of this study, delays for switching and routing decisions are ignored and only the delay in the physical channels is considered. The message generation interval has exponential distribution. Three different destination distribution patterns are considered; they are uniform random, hotspot, and Rentian [12]. Rentian traffic pattern is used because it is shown in [29] that many parallel applications employ a communication pattern that follows the Rent's rule. We model the Communication Probability Distribution (CPD), the probability that a node sends a message to another node at a given distance, as stated in [6]. For that we choose the Rent's exponent to be 0.7. For the hotspot traffic pattern, 10% of messages are destined for the hotspot node. Physical channel delay is one cycle and message length is chosen to be 64 flits. The networks use wormhole flow control [17].

We have chosen two different network sizes for the simulations. Table 3.2 summarizes the number of nodes, diameter, and average distance for the networks used in this study. The small-size pair consists of a 5-ary 4-cube (a four-dimensional torus) paired with $G^{(2)}_{3+4i}$. Each network has 625 nodes. For the large-size pair we compare a 12-ary 4-cube and a $G^{(2)}_{8+9i}$ which have 20736 and 21025 nodes, respectively.

Figures 3.8 shows the average message latency of deterministic routing as a function of the message generation rate for a uniform random traffic pattern. These results show our routing algorithm outperforms the dimension order routing algorithm for a torus with the same number of nodes and the same node degree.

Figures 3.9 shows the same result for the hotspot traffic pattern. The network saturates at significantly lower message generation rate per cycle using hotspot traffic pattern as compared to the uniform traffic pattern. Our routing algorithm

Table 3.2: The diameter and the average distance for the networks used in this study.

Network	# of nodes	Diameter	Avg distance
$G^{(2)}_{3+4i}$	625	6	4.48
$T_{5\times5\times5\times5}$	625	8	4.8
$G^{(2)}_{8+9i}$	21025	16	11.25
$T_{12\times12\times12\times12}$	20736	24	12

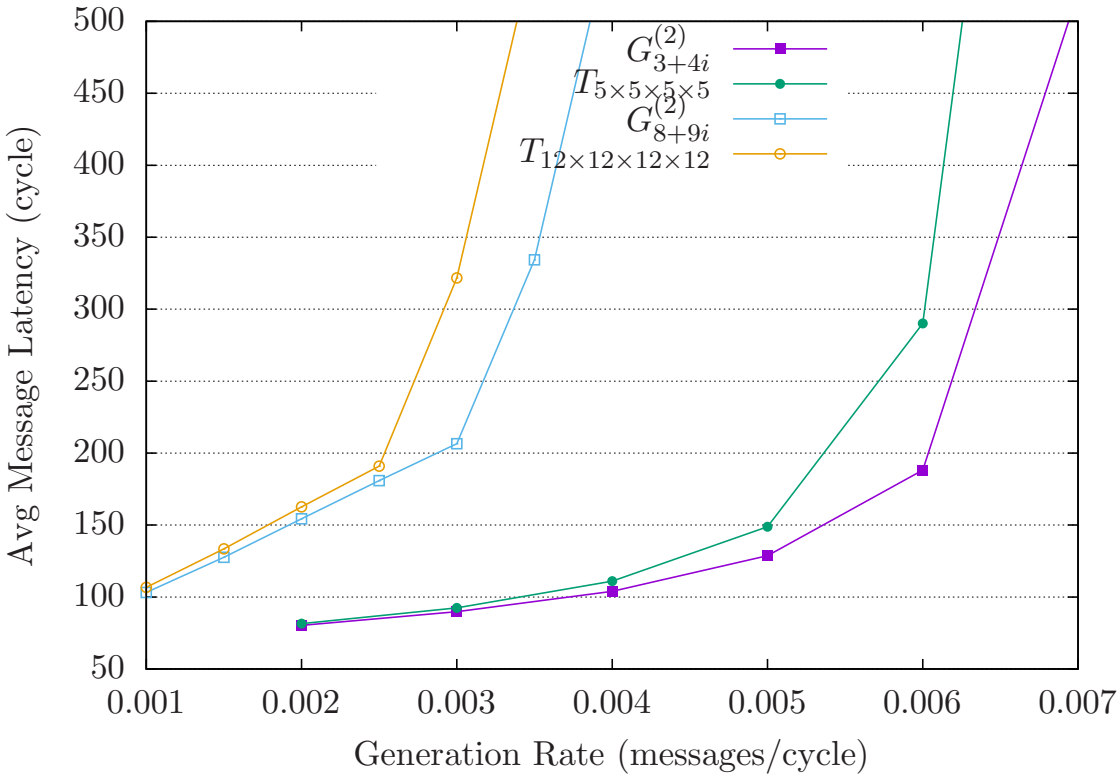


Figure 3.8: Deterministic routing using uniform random traffic

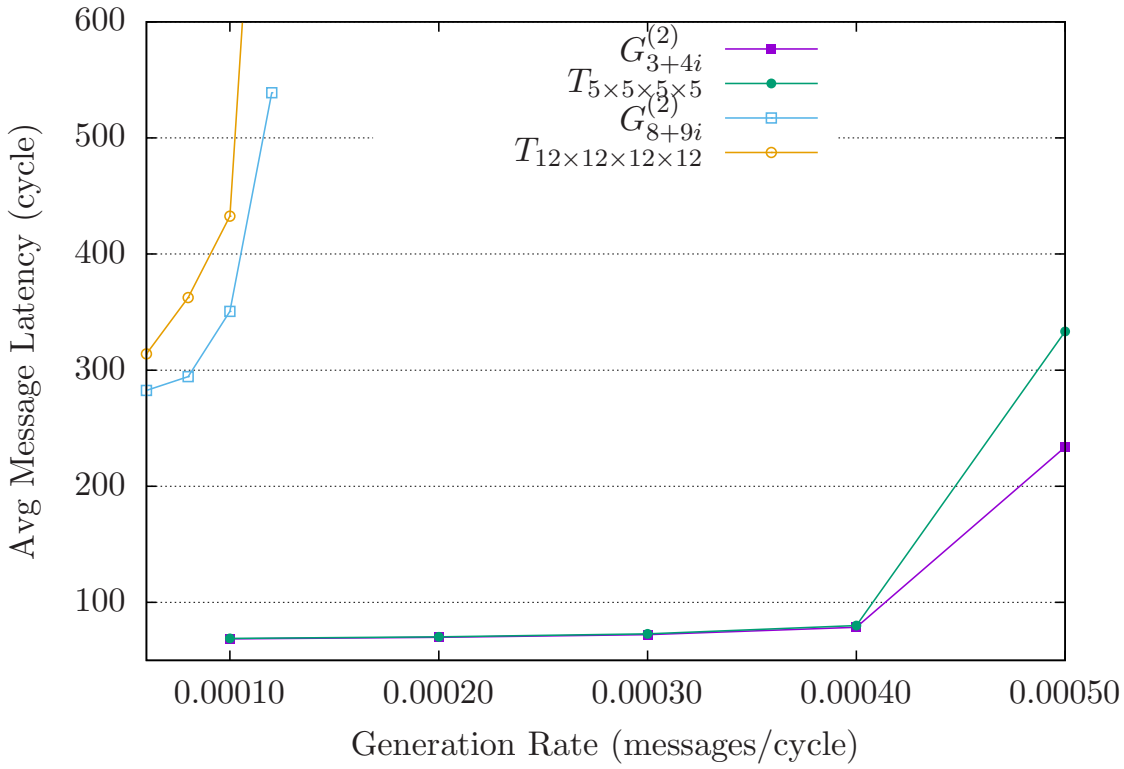


Figure 3.9: Deterministic routing using hotspot traffic

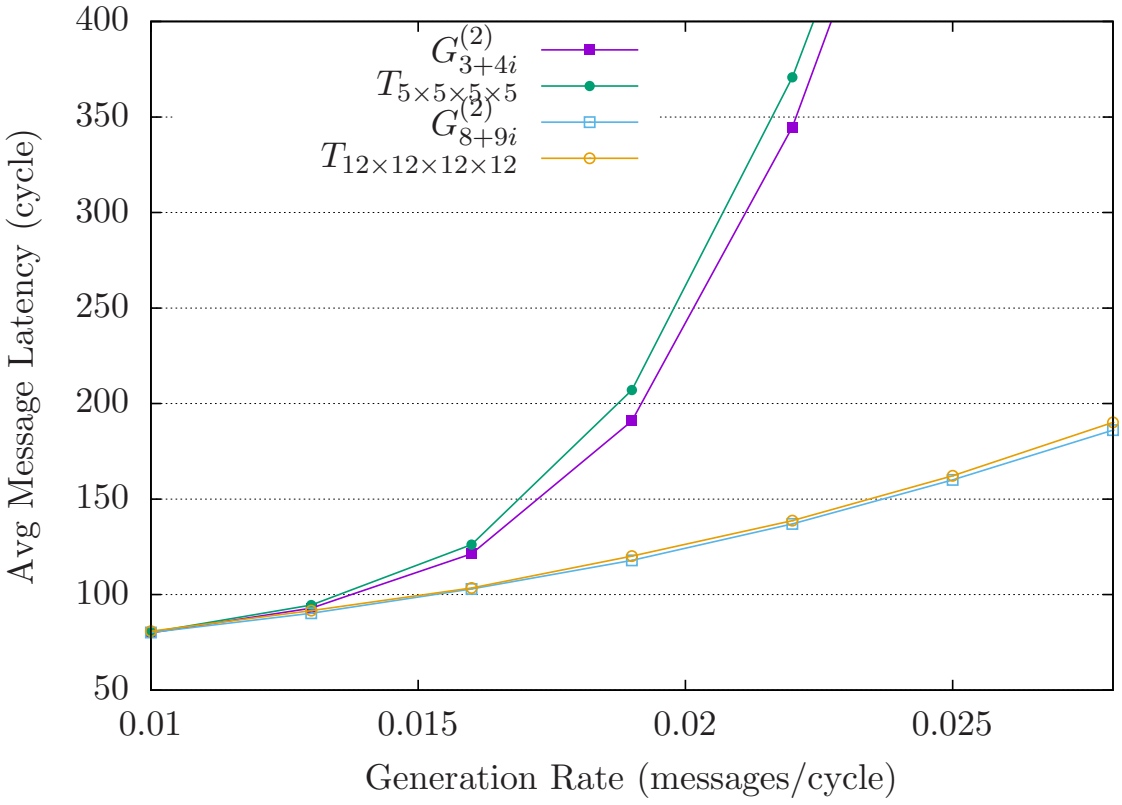


Figure 3.10: Deterministic routing using Rentian traffic

continues to outperform the routing algorithm of tori with the same number of nodes and the same node-degree.

Figure 3.10 shows the result for Rentian traffic pattern. Unlike previous two figures, our routing algorithm performs slightly better than the routing algorithm of tori in the small-size networks. The two routing algorithms perform almost the same in the large-size networks. Another interesting result is that the average message latency for the large-size networks is smaller than that of the small-size networks. This is due to the fact that as the network size increases the probability of communication between nodes at farther distances decreases.

3.7 Broadcasting

This section presents a one-to-all broadcasting method for n-dimensional Gaussian networks. In this communication pattern a node sends its message to all other nodes in the network. Broadcasting is frequently used in combinatorial and numerical problems [5, 33], and developing efficient algorithms for this communication pattern provides a design of efficient parallel solutions to these problems. The broadcasting algorithm presented in this section assumes the use of multiple I/O routers. That is, a node can send multiple messages simultaneously through all of its links.

An optimal one-to-all broadcasting algorithm for one-dimensional Gaussian networks was developed in [36]. We will explicitly describe the algorithm [38] for dense Gaussian networks below, and then explain how the one-dimensional algorithms can be iteratively applied to get an optimal algorithm for the n-dimensional case. A similar method works in non-dense networks.

For broadcasting in the dense network $G_{k+(k+1)i}$ a spanning tree is constructed as in Figure 3.11 in k steps that indicate broadcasting stages. All links incident to the node 0 are included in the tree, thereby connecting 0 to all four of its neighbors, $1, -i, -1, i$. These four neighbors form four separate right-angled quadrants, which we label as $SE, SW, NW,$ and NE , respectively. The remainder of our explanation focuses on the construction of the subtree in the SE quadrant since a similar

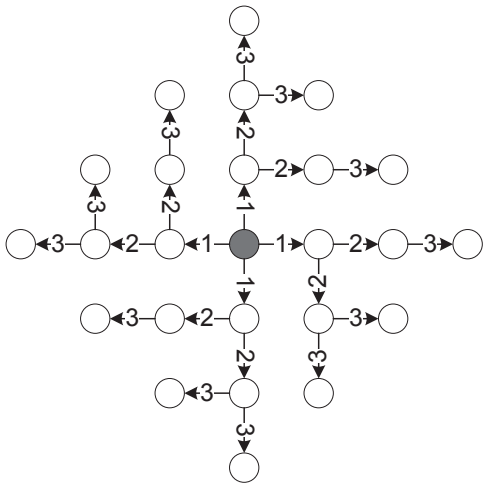


Figure 3.11: The spanning tree for G_{3+4i}.

process yields the subtrees in other quadrants. In the second stage of construction within the SE quadrant, the $+1$ and $-i$ edges from node 1 are added, giving access to the nodes 2 and $1-i$, respectively. At the third stage, the $+1$ and $-i$ edges of $1-i$ are added as well as the $+1$ edges from node 2. In general at the jth stage (for $j = 2, 3, \ldots, k$) the $+1$ and $-i$ edges from $1-(j-2)i$ are added as well as the $+1$ edges in any rows of the constructed spanning tree so far. Performing this procedure in each quadrant results in a spanning tree in which each node is within a distance of k from 0. The broadcasting from node 0 is performed in the stages indicated by this construction.

The broadcasting algorithm for n-dimensional networks of the form $G^{(n)}_{k+(k+1)i}$ is performed along a spanning tree similarly to the one-dimensional tree. Since an n-dimensional Gaussian network is node-symmetric, we may assume the node 0 is the center of broadcasting. The broadcasting is done in nk stages divided into n rounds of k stages each. The method adapts the idea in [48] to Gaussian networks."

In the first round of broadcasting, the broadcast message propagates in k stages as follows. At the first stage, node $(0, 0, \ldots, 0)$ sends the broadcast message to nodes $(1, 0, \ldots, 0), (-1, 0, \ldots, 0), (i, 0, \ldots, 0)$, and $(-i, 0, \ldots, 0)$. Then, at the next

stages every one of these nodes broadcasts in their corresponding quadrant in a manner similar to the process explained for the one-dimensional Gaussian network. That is, by the end of first round, all nodes $(a_{n-1}, 0, 0, \ldots, 0)$ have received the broadcast message. Figure 3.12 illustrates how the broadcast message propagates in the first round of broadcasting algorithm in $G^{(2)}_{3+4i}$.

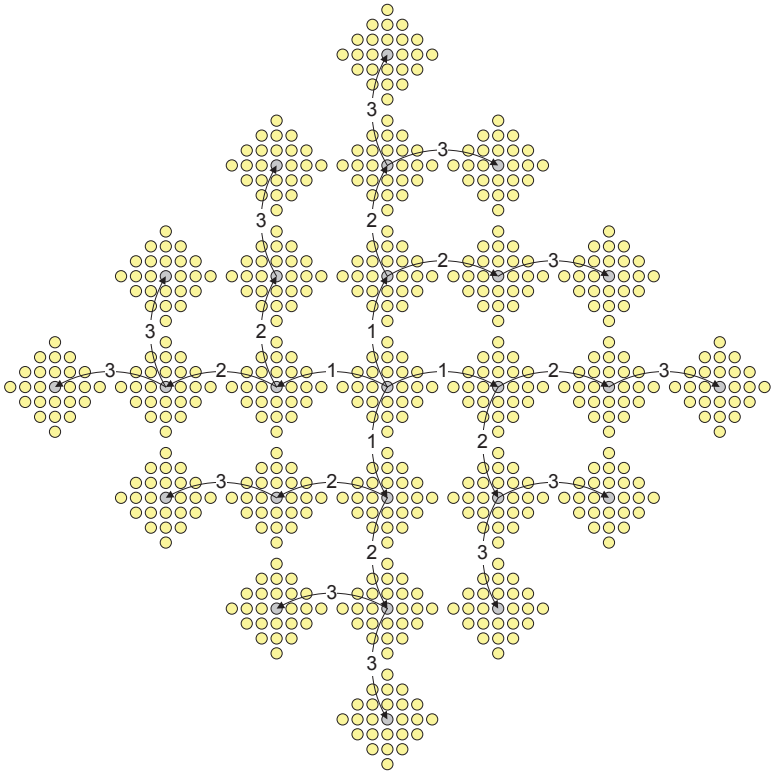


Figure 3.12: The spanning tree for the first round of broadcast algorithm in $G^{(2)}_{3+4i}$.

Similarly, by the end of second round, all nodes $(a_{n-1}, a_{n-2}, 0, \ldots, 0)$ have received the broadcast message in the k stages of the second round. This process can be continued. In the last round of broadcasting, the broadcast message has arrived at the centers $(a_{n-1}, a_{n-2}, \ldots, a_1, 0)$ of all the one-dimensional Gaussian networks in the lowest level of $G^{(n)}_{\alpha}$. Then broadcasting is continued similarly to broadcasting for one-dimensional Gaussian networks. This algorithm is optimal because no node receives duplicate messages and the number of steps in the algorithm is the

diameter of the network.

3.8 Embedding Multi-dimensional Tori on $G_\alpha^{(n)}$

One of the important problems in parallel processing is to determine *guest networks* that can be embedded onto a given network (called the *host network*). Then any software systems and algorithms developed for the guest networks can be transported to the host network. In this section for a given host network $G_\alpha^{(n)}$, $\alpha = a + bi$ where $\gcd(a, b) = 1$ and $n = 2^r, r \in \mathbb{Z}$, it is shown how to embed two edge-disjoint $(a^2 + b^2)$-ary n-cube, four edge-disjoint $(a^2 + b^2)$-ary $n/2$-cube, eight edge-disjoint $(a^2 + b^2)$-ary $n/2^2$-cube, etc. on an n-dimensional Gaussian network. In particular, it is shown how to decompose $G_\alpha^{(n)}$ into $2n$ edge-disjoint Hamiltonian cycles using some Lee distance Gray codes.

In the one-dimensional case it is known [24] that the two cycles formed by consistently adding either 1 or i to 0 contain all nodes of G_{a+bi} when $\gcd(a, b) = 1$, and so are two edge-disjoint Hamiltonian cycles. This establishes a one-to-one correspondence between the nodes in G_α and the nodes of the cycle of integers modulo $N = a^2 + b^2$ [36]. Since the node addresses in the n-dimensional Gaussian network are n-tuples, this gives a one-to-one correspondence between the nodes in $G_\alpha^{(n)}$ and nodes in the n-dimensional torus with radix N.

Example 3.6. *Consider G_{1+2i} which has $1^2+2^2 = 5$ nodes. The node address set is $\{0, 1, -1, i, -i\}$ and G_{1+2i} is a complete graph with 5 nodes. There is a one-to-one correspondence between $Z_5 = \{0, 1, 2, 3, 4\}$ and the node set $\{0, 1, -1, i, -i\}$ because $0 \mod (1+2i) = 0$, $1 \mod (1+2i) = 1$, $2 \mod (1+2i) = i$, $3 \mod (1+2i) = -i$, and $4 \mod (1+2i) = -1$. In this network there are two edge-disjoint Hamiltonian cycles. The first is*

$$H_1 = \langle 0, 1, 2, 3, 4\rangle = \langle 0, 1, i, -i, -1\rangle$$

and the second is

$$H_2 = \langle 0, i, 2i, 3i, 4i\rangle = \langle 0, i, -1, 1, -i\rangle.$$

3.8.1 Hamiltonian Property of $G_\alpha^{(n)}$ for any $n \geq 1$

First we show that $G_\alpha^{(n)}$ is Hamiltonian. In what follows, we will denote the node addresses in $G_\alpha^{(n)}$ as elements in $\mathbb{Z}_\alpha^n$ and hence by the above discussion in $\mathbb{Z}_N^n$. We will see that the Gray code given in [13] and [7] described below can be used to show that $G_\alpha^{(n)}$ is Hamiltonian. Note that in a Gray code over $\mathbb{Z}_N$, every two consecutive vectors and also the last and the first vectors differ in one position by $\pm 1 \mod N$.

Assume $\mathbb{Z}_N^n$ is ordered in the radix-N number system. The Gray code mapping $f : \mathbb{Z}_N^n \to \mathbb{Z}_N^n$ as given by [13] and [7] is

$$f(a_{n-1}a_{n-2}\dots a_0) = (g_{n-1}g_{n-2}\dots g_0)$$

where $g_{n-1} = a_{n-1}$ and for $j = n-2, n-3, \dots, 0$, $g_j = (a_j - a_{j+1}) \mod N$.

Example 3.7. *Since $\mathcal{N}(1+2i) = 5$ then $N = 5$. There are therefore 25 nodes in the network $G_{1+2i}^{(2)}$. The Gray code sequence over $\mathbb{Z}_5^2$ is*

$$\begin{aligned} H = \langle &00, 01, 02, 03, 04, 14, 10, 11, 12, 13, 23, 24, 20, 21, 22, \\ &32, 33, 34, 30, 31, 41, 42, 43, 44, 40\rangle \\ = \langle &00, 01, 0i, 0-i, 0-1, \dots, -10\rangle \end{aligned}$$

and this forms one Hamiltonian cycle. Please refer to Figure 3.13. The solid lines in the first part of the network gives this Hamiltonian cycle.

Note that iH also forms another Hamiltonian cycle. Furthermore, H and iH are edge-disjoint because in H two adjacent nodes differ by ± 1 and in iH they differ by $\pm i$.

3.8.2 Edge-Disjoint Hamiltonian Cycles and Tori in $G_\alpha^{(2)}$

In this subsection, we describe the decomposition technique for a $G_\alpha^{(2)}$. The next subsection describes the method for a $G_\alpha^{(n)}$ where $n = 2^r, r > 1$.

Many Hamiltonian decomposition results obtained for the torus as given in [4] can be directly applied to the n-dimensional Gaussian network. For $n = 2^r$, these results can be easily described in terms of Gray codes.

As recalled above, G_{a+bi} can be decomposed into two edge-disjoint Hamiltonian cycles by consistently adding 1 or i to the starting node $N = a^2 + b^2$ times. For example, G_{1+2i} has two edge disjoint Hamiltonian cycles given by $H_1 = \langle 0, 1, 2, 3, 4\rangle$ and $H_2 = \langle 0, i, 2i, 3i, 4i\rangle$. In what follows, this decomposition is written as

$$G_\alpha = H_1 \oplus H_2.$$

Now consider the two dimensional Gaussian network $G_\alpha^{(2)}$. The general idea of how to obtain two edge-disjoint tori and four edge-disjoint Hamiltonian cycles in this network is described below.

Since each 1-dimensional factor can be decomposed into two edge-disjoint Hamiltonian cycles of length N, $G_\alpha^{(2)}$ can be decomposed as

$$G_\alpha^{(2)} = G_\alpha \otimes G_\alpha = (H_1^N \oplus H_2^N) \otimes (H_3^N \oplus H_4^N).$$

This last term equals $(H_1^N \otimes H_3^N) \oplus (H_2^N \otimes H_4^N)$ due to the distribution property of the operands, where each summand is a torus of size $N \times N$. Therefore, these 2-dimensional Gaussian networks can be decomposed into two edge disjoint $N \times N$ tori. In [4] (also see below), it is shown how to obtain two edge-disjoint Hamiltonian cycles in a torus network. Using this scheme for each of the above tori, we have decomposed the 2-dimensional network $G_\alpha^{(2)}$ into four edge-disjoint Hamiltonian cycles. This is illustrated in the next example and its accompanying Figure 3.13.

Example 3.8. *Let* $\alpha = 1 + 2i$. *Then:*

$$\begin{aligned} G^{(2)}_{1+2i} &= G_{1+2i} \otimes G_{1+2i} \\ &= (\langle 0,1,2,3,4\rangle \oplus \langle 0,i,2i,3i,4i\rangle) \\ &\otimes (\langle 0,1,2,3,4\rangle \oplus \langle 0,i,2i,3i,4i\rangle) \\ &= (\langle 0,1,2,3,4\rangle \otimes \langle 0,1,2,3,4\rangle) \\ &\oplus (\langle 0,i,2i,3i,4i\rangle \otimes \langle 0,i,2i,3i,4i\rangle) \\ &= T_{5\times 5} \oplus T_{5\times 5} \\ &= (H'_1 \oplus H'_2) \oplus (H'_3 \oplus H'_4) \end{aligned}$$

$\langle 0,1,2,3,4\rangle \otimes \langle 0,1,2,3,4\rangle$ *and* $\langle 0,i,2i,3i,4i\rangle \otimes \langle 0,i,2i,3i,4i\rangle$ *are two edge-disjoint tori as shown in Figure 3.13. From these two tori, we can obtain four edge-disjoint Hamiltonian cycles, each of length* 25. *In the figure, the solid lines and dashed lines show these four Hamiltonian cycles.*

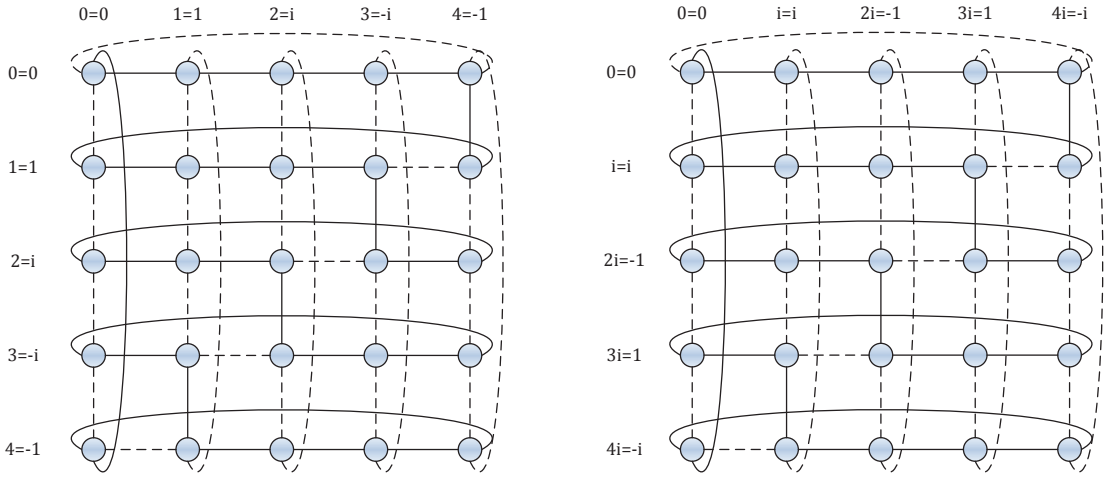


Figure 3.13: Two edge-disjoint tori $T^1_{5\times 5} \oplus T^2_{5\times 5}$ generating four edge-disjoint Hamiltonian cycles in $G^{(2)}_{1+2i}$.

3.8.3 Gray Codes and Edge-Disjoint Hamiltonian Cycles in $G_\alpha^{(n)}$, where $n = 2^r, r > 1$

So far, we have shown that there exist two edge-disjoint tori and four edge-disjoint Hamiltonian cycles in the degree 8 network $G_\alpha^{(2)}$. In this section we describe these in terms of the Gray codes given in [4]. This will facilitate the generation of edge-disjoint tori in $G_\alpha^{(2)}$ as well as in $G_\alpha^{(n)}$ where $n = 2^r$. First, we describe a basic Gray code in a recursive way and then show how to modify this Gray code to generate all edge-disjoint Hamiltonian cycles in $G_\alpha^{(n)}$. Then, in the next section it is shown how to obtain edge-disjoint tori.

Let $G_N^{(1)} = (a_0, a_1, \ldots, a_{N-1})$ with $a_i \in \mathbb{Z}_N$ and all a_i's distinct where $N = a^2 + b^2$ is the number of nodes in the Gaussian network generated by $\alpha = a + bi$. Then $G_N^{(1)}$ is a Gray code with one digit. Let $G_N^{(1),t}$ be the cyclic t-shift of $G_N^{(1)}$. For example:

$$\begin{aligned}
G_N^{(1),0} &= G_N^{(1)} = (a_0, a_1, \ldots, a_{N-1}) \\
G_N^{(1),1} &= (a_{N-1}, a_0, a_1, \ldots, a_{N-2}) \\
G_N^{(1),2} &= (a_{N-2}, a_{N-1}, a_0, a_1, \ldots) \\
&\vdots \\
G_N^{(1),j} &= (a_{N-j}, a_{N-j+1}, \ldots, a_{N-1}, a_0, a_1, \ldots)
\end{aligned}$$

Define $G_N^{(2m)}$ as follows:

$$\begin{aligned}
G_N^{(2m)} &= G_N^{(m)} \otimes G_N^{(m)} \\
&= \{A_j G_N^{(m),j} | A_j \text{ is the } j\text{-th word in the } G_N^{(m)}\} , \\
&\quad \text{for } j = 0, 1, 2, \ldots, N-1\}
\end{aligned}$$

Example 3.9. *Let $\alpha = 1 + 2i$. Then $N = 5$ and we have:*

$$G_5^{(1),0} = G_5^{(1)} = (0, 1, 2, 3, 4)$$
$$G_5^{(1),1} = (4, 0, 1, 2, 3)$$
$$G_5^{(1),2} = (3, 4, 0, 1, 2)$$
$$G_5^{(1),3} = (2, 3, 4, 0, 1)$$
$$G_5^{(1),4} = (1, 2, 3, 4, 0)$$
$$G_5^{(2)} = G_5^{(1)} \otimes G_5^{(1)} \text{ as shown in Table 3.3}$$
$$G_5^{(4)} = G_5^{(2)} \otimes G_5^{(2)} \text{ as shown in Table 3.4.}$$

Table 3.3: Gray Codes in $G_\alpha^{(2)}$ where $\alpha = 1 + 2i$.

$0G_5^{(1),0}$	$1G_5^{(1),1}$	$2G_5^{(1),2}$	$3G_5^{(1),3}$	$4G_5^{(1),4}$
00	14	23	32	41
01	10	24	33	42
02	11	20	34	43
03	12	21	30	44
04	13	22	31	40

The following Claim proved in [4] shows that the $G_N^{(2m)}$ defined above is actually a Gray code.

Claim 3.10. *$G_N^{(n)}$ forms a Gray code over $\mathbb{Z}_N^n$ for n a power of 2, i.e, $n = 2^r$.*

Next we show that all other edge-disjoint Hamiltonian cycles in $G_\alpha^{(n)}$ can be obtained by some form of permutation and multiplication over the digits of $G_N^{(n)}$. First, we define these permutations.

For $0 \leq j \leq 2^n - 1$, let $j = (j_{n-1}\ j_{n-2}\ \ldots\ j_0)$ be the binary representation of the integer j. Let $G_N^{(n=2^r)}$ be the Gray code with column digits as $(e_{n-1}\ e_{n-2}\ \ldots\ e_0)$. Then $P_j(G_N^{(n=2^r)})$ is the permutation of the column vectors of $G_N^{(n=2^r)}$, for $j = 0, 1, 2, \ldots, 2^r - 1$ and these permutations are defined as follows.

Table 3.4: Gray Codes in $G_\alpha^{(4)} = G_{5,2} \otimes G_{5,2}$ where $\alpha = 1 + 2i$.

$00\, G_5^{(2),0}$	$01\, G_5^{(2),1}$	...	$44\, G_5^{(2),23}$	$40\, G_5^{(2),24}$
00 00	01 40	...	44 02	40 01
00 01	01 00	...	44 03	40 02
00 02	01 01	...	44 04	40 03
00 03	01 02	...	44 14	40 04
00 04	01 03	...	44 10	40 14
00 14	01 04	...	44 11	40 10
00 10	01 14	...	44 12	40 11
00 11	01 10	...	44 13	40 12
00 12	01 11	...	44 23	40 13
00 13	01 12	...	44 24	40 23
00 23	01 13	...	44 20	40 24
00 24	01 23	...	44 21	40 20
00 20	01 24	...	44 22	40 21
00 21	01 20	...	44 32	40 22
00 22	01 21	...	44 33	40 32
00 32	01 22	...	44 34	40 33
00 33	01 32	...	44 30	40 34
00 34	01 33	...	44 31	40 30
00 30	01 34	...	44 41	40 31
00 31	01 30	...	44 42	40 41
00 41	01 31	...	44 43	40 42
00 42	01 41	...	44 44	40 43
00 43	01 42	...	44 40	40 44
00 44	01 43	...	44 00	40 40
00 40	01 44	...	44 01	40 00

For $j = (j_{n-1}\ j_{n-2}\ \ldots\ j_0)$, if $j_k = 1, k = 0, 1, \ldots, n-1$ then permute the least 0-th 2^k elements of $G_N^{(n=2^r)}$ with next (1-st) 2^k elements, the second 2^k elements with the third 2^k elements, and so on. For example, let $P_0(G_N^{(2^3)}) = P_{000}(G_N^{(2^3)}) = (e_7, e_6, e_5, e_4, e_3, e_2, e_1, e_0)$. Then

$$
\begin{aligned}
P_1(G_N^{(2^3)}) &= P_{001}(G_N^{(2^3)}) = (\underline{e_6}, \underline{e_7}, \underline{e_4}, \underline{e_5}, \underline{e_2}, \underline{e_3}, \underline{e_0}, \underline{e_1}) \\
P_2(G_N^{(2^3)}) &= P_{010}(G_N^{(2^3)}) = (\underline{e_5, e_4}, \underline{e_7, e_6}, \underline{e_1, e_0}, \underline{e_3, e_2}) \\
P_3(G_N^{(2^3)}) &= P_{011}(G_N^{(2^3)}) = (\underline{\underline{e_4, e_5}}, \underline{\underline{e_6, e_7}}, \underline{\underline{e_0, e_1}}, \underline{\underline{e_2, e_3}}) \\
P_4(G_N^{(2^3)}) &= P_{100}(G_N^{(2^3)}) = (\underline{e_3, e_2, e_1, e_0}, \underline{e_7, e_6, e_5, e_4}) \\
P_5(G_N^{(2^3)}) &= P_{101}(G_N^{(2^3)}) = (\underline{\underline{e_2, e_3}, \underline{e_0, e_1}}, \underline{\underline{e_6, e_7}, \underline{e_4, e_5}}) \\
P_6(G_N^{(2^3)}) &= P_{110}(G_N^{(2^3)}) = (\underline{\underline{e_1, e_0}, \underline{e_3, e_2}}, \underline{\underline{e_5, e_4}, \underline{e_7, e_6}}) \\
P_7(G_N^{(2^3)}) &= P_{111}(G_N^{(2^3)}) = (\underline{\underline{\underline{e_0, e_1}, \underline{e_2, e_3}}}, \underline{\underline{\underline{e_4, e_5}, \underline{e_6, e_7}}})
\end{aligned}
$$

Now it can be shown [4] if $G_N^{(n=2^r)}$ is the Gray code designed as in Claim 3.10, then

- the permutations $P_j(G_N^{(n=2^r)})$, for $j = 0, 1, 2, \ldots, 2^r - 1$ give $n = 2^r$ edge-disjoint Hamiltonian cycles in $G_\alpha^{(n)}$

Furthermore, it can be proved that

- multiplying each of $P_j(G_N^{(n=2^r)})$ by the complex number i gives another set of $n = 2^r$ edge-disjoint Hamiltonian cycles in $G_\alpha^{(n)}$.

In this way, $2n$ edge-disjoint Hamiltonian cycles are generated in $G_\alpha^{(n)}$ for $n = 2^r$.

3.8.4 Embedding Edge-disjoint Tori

So far we have shown how to obtain $2n$ edge-disjoint Hamiltonian cycles in $G_\alpha^{(n)}$ where $n = 2^r$, and $\alpha = a + bi$ with $\gcd(a, b) = 1$. Now we describe how to embed various edge-disjoint tori on $G_\alpha^{(n)}$. First it is explained for two-dimensional

tori. Consider the two dimensional tori of size $\mathcal{N}(\alpha)^{n/2} \times \mathcal{N}(\alpha)^{n/2}$ whose row and column addresses correspond to one of the $n/2 = 2^{r-1}$ digit independent Gray codes mentioned above. (Two Gray codes are said to be independent if they generate edge-disjoint Hamiltonian cycles in the corresponding torus.) Since there are $2(n/2) = 2^r$ independent Gray codes we will have 2^r two-dimensional tori. These tori are edge-disjoint. This is because, if two nodes are adjacent in a row (respectively in a column) of a given torus, then they cannot be adjacent in a row (respectively in a column) of any other torus since the rows and columns to be addressed correspond to independent Gray codes. In a similar fashion, we can obtain $2(n/2^2) = 2^{r-1}$ edge-disjoint 4-dimensional tori by using the $n/4$ digit independent Gray codes as the addresses along each dimension; 2^{r-2} edge-disjoint 8-dimensional tori by using $n/8$ digit independent Gray codes as the addresses along each dimension, etc.

3.9 Higher Dimensional EJ Networks

The theory and topological properties developed for n-dimensional Gaussian networks can easily be extended to first define and then to develop topological properties of n-dimensional EJ networks. In [37], the authors have presented the original one-dimensional EJ network and in [24] many topological properties of this network are given.

The hexagonal network is a special case of an EJ network [2]. There have been several examples of higher dimensional hexagonal mesh networks described in [9, 21, 26]. The basis for all these networks is a hexagonal mesh, and they are not node-symmetric.

Let $\omega = a + b\rho$ where $\rho = \frac{1+\sqrt{3}i}{2}$ be a given EJ integer. The norm is $\mathcal{N}(\omega) = a^2 + b^2 + ab$. In this network E_ω, generated by ω, the number of nodes is $\mathcal{N}(\omega)$ and each node can be addressed using EJ integers in $\mathbb{Z}_\omega$. Two nodes $\alpha_1, \alpha_2 \in \mathbb{Z}_\omega$are adjacent if and only if they differ by $\pm 1, \pm\rho$ or $\pm\rho^2$. Then, E_ω is a regular degree six network with a^2+b^2+ab nodes. Now we describe the n-dimensional EJ network $E_\omega^{(n)}$.

Definition 3.11. *An n-dimensional EJ network $E_\omega^{(n)}$ generated by $\omega = a + b\rho$ has $\mathcal{N}(\omega)^n = (a^2 + b^2 + ab)^n$ nodes, each node is addressed using the EJ integers in $\mathbb{Z}_\omega^n$ and two nodes $\beta_1, \beta_2 \in \mathbb{Z}_\omega^n$ are connected by an edge if and only if they differ in one position by $\pm 1, \pm\rho$ or $\pm\rho^2$.*

Thus, the degree of these networks is $6n$. It can be easily seen that these networks satisfy the Cayley graph property with $6n$ generators. These generators have 0's in all but one position, where the non-zero element can be $1, -1, \rho, -\rho, \rho^2$ or $-\rho^2$. This implies the n-dimensional EJ networks are node-symmetric.

The routing and broadcasting algorithms for these networks can be done using the dimensional order routing (broadcasting) algorithm similar to the ones described in Section 3.4 (Section 3.7) for n-dimensional Gaussian networks. In addition, the distance distribution of these networks can also be obtained using the recurrence equation method described in Section 3.5.

Another way of defining $E_\omega^{(n)}$ is using the concept of the cross product of graphs, i.e.

$$E_\omega^{(n)} = \overbrace{E_\omega \otimes E_\omega \otimes \ldots \otimes E_\omega}^{n \text{ times}}.$$

Given $\omega = a+b\rho$ where $\gcd(a, b) = 1$, it is shown in [24], E_ω can be decomposed into three edge-disjoint Hamiltonian cycles. These three edge-disjoint Hamiltonian cycles can be obtained by traversing along 1,ρ, and ρ^2 directions. Suppose $n = 2^r$ for $r \in \mathbb{Z}$. Similar to the techniques given in Section 3.8, we can embed three edge-disjoint $(a^2 + b^2 + ab)$-ary n-cube on $E_\omega^{(n)}$, or six edge-disjoint $(a^2 + b^2 + ab)$-ary $n/2$-cube, or 12 edge-disjoint (a^2+b^2+ab)-ary $n/2^2$-cube, and so on. In particular, we can generate $3 \times 2^r = 3n$ edge-disjoint Hamiltonian cycles in $E_\omega^{(n)}$. The first n edge-disjoint Hamiltonian cycles can be generated by the Lee distance Gray codes described in Section 3.8. Then, multiplying these Gray codes by ρ and ρ^2 we can get another set of $2n$ Gray codes and these $3n$ Gray codes correspond to $3n$ edge-disjoint Hamiltonian cycles in $E_\omega^{(n)}$ where $\omega = a + b\rho$.

3.10 Conclusion

In this chapter, we have introduced higher dimensional Gaussian networks and have shown that the diameter and the average distance of the higher dimensional Gaussian network are less than those of a multidimensional torus with the same number of nodes and degree. An optimal one-to-one routing algorithm has been developed. Simulation results show that our given routing algorithm does outperform the usual routing algorithm for tori in terms of average message latency.

We have been able to extend many properties of one-dimensional Gaussian networks to the higher dimensional setting, including finding the distance distribution and an optimal one-to-all broadcasting algorithm. In particular, when $n = 2^r$ we have generated $2n$ edge-disjoint Hamiltonian cycles in n-dimensional Gaussian networks $G^{(n)}_{a+bi}$ when $\gcd(a, b) = 1$. We have also shown how to embed multi-dimensional edge-disjoint torus networks on this network.

We have also briefly shown that the development of the theory in this chapter can easily be extended to the higher dimensional EJ networks.

Chapter 4: Degree-three Pruned Gaussian Networks

4.1 Introduction

Gaussian networks [38] were proposed as an alternative to toroidal networks. These networks are shown to have advantage over the toroidal networks because they have a smaller diameter. A smaller diameter and smaller average distance implies a smaller average message latency, the average time of arrival and departure of messages in the network. It is shown in [36] that a 2D torus is a special case of a Gaussian network. These networks are regular of degree four. Smaller degree interconnection networks are sometimes more desirable, especially in on-chip networks. The smaller the degree of a node, the lower the cost of implementation.

Some attempts have been made to introduce degree-three networks to obtain networks with smaller degree than degree four. Honeycomb mesh networks [44] are a well-known class of degree-three networks. Adding appropriate wraparound edges to a honeycomb mesh yields a honeycomb torus. Honeycomb torus networks are known to cost less than 2D toroidal networks where the cost of a network is defined as the product of the degree and the diameter of the network [44].

Parhami and Kwai in [41] and also Xiao and Parhami in [47] showed that a honeycomb torus can be obtained by pruning a 2D torus. We apply the analogous pruning technique to Gaussian networks in order to construct degree-three pruned Gaussian networks. Just as Gaussian networks are a generalization of 2D square torus networks, we show that degree-three pruned Gaussian networks are a generalization of honeycomb square torus networks. In addition, in some situations the resulting degree-three pruned Gaussian network has a smaller diameter than the corresponding honeycomb torus with an equivalent number of nodes.

The rest of this chapter is organized as follows. In Section 4.2 some basic definitions and mathematical terms are explained. Section 4.3 explains the honey-

comb network. Next, Section 4.4 briefly reviews important properties of Gaussian networks that are relevant to the rest of the chapter. The definition and some basic topological properties of degree-three pruned Gaussian networks are given in Section 4.5. The summarizing conclusions and future research directions are given in Section 4.6.

4.2 Definitions

In this chapter we briefly review some of the terminology and mathematical terms that are used throughout the rest of this section. An interconnection network is an undirected graph. Thus, the following pair of terms are equivalent: node/vertex, edge/link, graph/network. We use node, edge, and network throughout the rest of this manuscript.

Definition 4.1. *Let $G = (V, E)$ be a graph. Then $V(G)$ represents the set of nodes and $E(G)$ represents the set of edges in G. The number of edges incident to a node is defined to be the degree of the node.*

Definition 4.2. *A simple graph is an undirected, unweighted graph with no loops and no multi-edges. A simple graph could be connected or disconnected.*

Definition 4.3. *Let G and H be two simple graphs. A graph isomorphism from G to H is a bijection $f : V(G) \to V(H)$ such that for any two vertices u and v in $V(G)$, $uv \in E(G)$ if and only if $f(u)f(v) \in E(H)$. That is, u and v are adjacent in G if and only if $f(u)$ and $f(v)$ are adjacent in H. The two graphs G and H are said to be isomorphic if and only if there exists an isomorphism between them.*

Definition 4.4. *An automorphism of a graph G is an isomorphism of the graph with itself. That is, there is a bijection from $V(G)$ to $V(G)$ that maps every pair of edges in $E(G)$ to $E(G)$.*

Definition 4.5. *A graph is* node-symmetric *if for every pair of nodes u and v there exists an automorphism mapping u to v.*

Definition 4.6. *A set* Γ *is said to be a* group *under the binary operation* $\otimes$ *if it satisfies the following properties.*

- *Closure:* $\forall x, y \in \Gamma, x \otimes y \in \Gamma$.
- *Identity:* $\exists 1 \in \Gamma$ *such that* $\forall x \in \Gamma : x \otimes 1 = 1 \otimes x = x$.
- *Associativity:* $\forall x, y, z \in \Gamma$ *then* $(x \otimes y) \otimes z = x \otimes (y \otimes z)$.
- *Inverse:* $\forall x \in \Gamma, \exists x^{-1} \in \Gamma$ *such that* $x \otimes x^{-1} = x^{-1} \otimes x = 1$.

Definition 4.7. *Let* S *be a subset of the group* Γ. *Then* S *is called a* generator set *of group* Γ *if every element of* Γ *can be obtained as a combination of finitely many elements of* S *or their inverses under the group operation.*

Definition 4.8. *Let* S *be a set of generators for a finite group* Γ *under the group operation* $\otimes$. *The* Cayley graph *relative to* Γ *and* S *is the graph in which the nodes are the elements of the group* Γ *and the edges are defined as follows. Two nodes* x *and* $y \in \Gamma$ *are adjacent if and only if there exists a generator* $s \in S$ *such that either* $y = x \otimes s$ *or* $y = s \otimes x$.

4.3 Honeycomb Networks

A honeycomb mesh network is built from hexagons in the hexagonal plane tessellation where the nodes and edges of the network are the nodes and edges of the hexagon. One way to build a honeycomb mesh network is to consider all the hexagonal tiles included in a fixed convex polygon. Then the sides of the polygon specify the borders of a honeycomb mesh network. The nodes on the borders are called the boundary nodes. Therefore, a honeycomb mesh network is a degree-three network where each node except for the boundary nodes is adjacent to three nearest neighbors 120 degrees apart. Three polygons (hexagon, rectangle, and rhombus) were used in [44] to define three types of honeycomb meshes namely the honeycomb hexagonal mesh (HHM or HM), the honeycomb rectangular mesh (HReM), and the honeycomb rhombic mesh (HRoM).

Honeycomb hexagonal meshes of size t (denoted by HM_t) were discussed thoroughly in [44]. Basically, one hexagon forms HM_1 and HM_2 is formed by placing 6 hexagons on the six sides of HM_1. In general, HM_t is formed by including the $6(t-1)$ hexagons surrounding the outer layer of HM_{t-1}. Figure 4.1(a) shows HM_3.

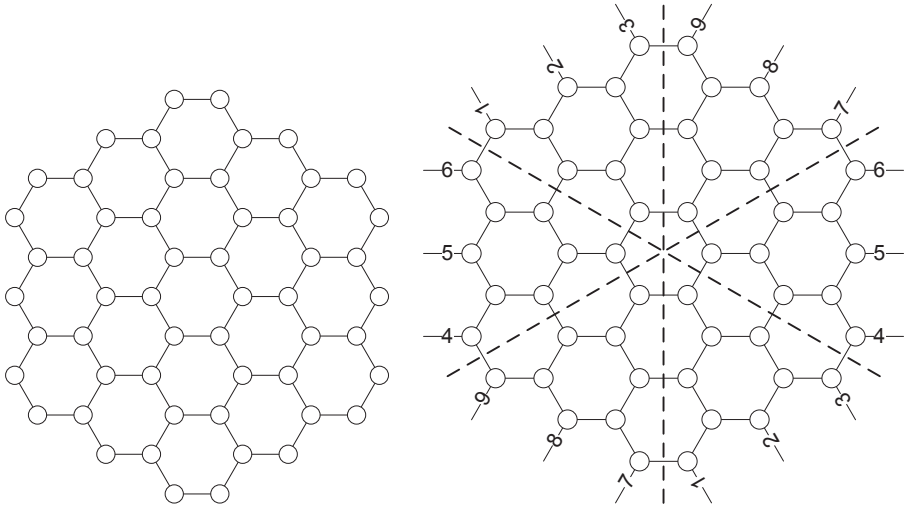
A honeycomb hexagonal torus of size t (HT_t) can be constructed from a honeycomb hexagonal mesh of size t (HM_t) by adding appropriate wraparound edges. Figure 4.1(b) gives the wraparound edges for HM_3. The same numbers represent the existence of a wraparound edge between two boundary nodes. Consider the three dashed lines in Figure 4.1(b) . These are the three axes used in [44] to address the nodes. Each pair of dashed lines divides the nodes into different sectors. In total, there are six sectors. Boundary nodes in one sector are adjacent to boundary nodes in the opposite sector. The wraparound edges are drawn by considering the mirror symmetric boundary nodes with respect to the third axis. For more information on these wraparounds refer to [44].

The number of nodes in HM_t and HT_t is $6t^2$. The number of edges, the diameter, and the bisection width of HM_t (HT_t) are $9t^2 - 3t$ ($9t^2$), $4t - 1$ ($2t$), and $0.82\sqrt{n}$ ($2.04\sqrt{n}$), respectively [44].

4.3.1 Honeycomb Rectangular Tori

Honeycomb rectangular tori were considered in [41] as networks with better topological properties than other honeycomb networks. These networks have a rectangular exterior shape and when redrawing them using a *brick drawing* [41] transformation, they resemble brick tori. A brick drawing of a network is another way of representing the same network in which the network is redrawn in such a way that the edges of the network become perpendicular to each other. (Refer to Figure 4.2.)

A honeycomb rectangular torus is denoted by $\mathrm{HReT}(m, n)$ where m and n are the sides of the exterior rectangle encompassing the honeycomb network. It is shown in [41] that $\mathrm{HReT}(m, n)$ is the pruned version of an $m \times 2n$ torus network; that is, $\mathrm{HReT}(m, n)$ is obtained by removing some of the edges from an $m \times 2n$ torus but keeping the same set of nodes (For one example refer to Figure 4.2).



(a) HM_3 (b) HT_3. Same numbers represent wraparound edges.

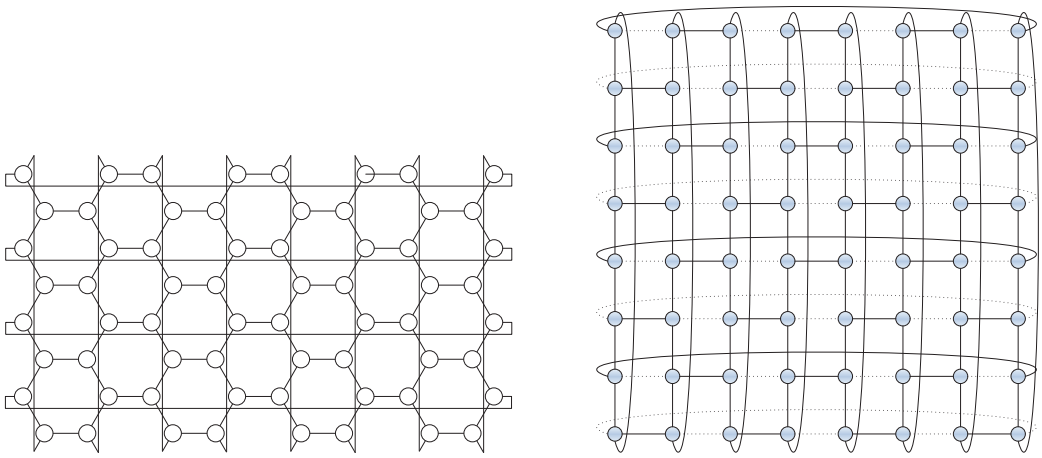
Figure 4.1: Honeycomb hexagonal mesh and torus of size 3.

For this reason, it is stated in [41] that a honeycomb rectangular torus has a straightforward VLSI layout. In addition, it is proved in [41] that honeycomb rectangular tori are Cayley graphs and thus node-symmetric. The authors in [41] have proposed an efficient routing algorithm for honeycomb rectangular tori.

The authors in [41, 44] studied the special case of HReT(m, n) where $m = 2n$. This special honeycomb torus has an square exterior shape and is denoted by HST_t or HReT$(t, t/2)$. The number of nodes in HST_t is t^2 and its diameter is t. Figure 4.2(a) shows the HReT(8,4) and its corresponding 8×8 pruned torus version is shown in Figure 4.2(b) in which the dotted lines represent the pruned (removed) edges.

4.4 A Review of Gaussian Networks

In this section we briefly recall Gaussian networks as defined earlier in Section 3.2 and present some more properties of these networks that will be useful in the rest of this chapter. More details can be found in [24, 36, 38].



(a) HReT(8,4) (b) 8×8 pruned torus

Figure 4.2: The honeycomb rectangular torus HReT(8,4) which is isomorphic to 8×8 pruned torus. The dotted lines represent the pruned edges in the 2D torus. The solid lines correspond to the brick drawing of HReT(8,4).

A Gaussian network G_α is generated by the complex number $\alpha = a + bi$ where a, b are positive integers. The nodes in G_α are the set of remainders upon division by α (the division as complex numbers). A node in G_α is addressed using a complex Gaussian integer in the form of $x + yi$ where the integers x and y are the real and imaginary parts of the Gaussian integer. Two nodes in G_α are adjacent if and only if they differ by ± 1 or $\pm i$ modulo α. A given node $x + yi \in G_\alpha$ is called an *even* node if $x + y$ is even, while $x + yi \in G_\alpha$ is called an *odd* node if $x + y$ is odd.

Definition 4.9. *A bipartite graph is a graph in which the nodes of the graph can be divided into two disjoint sets such that no two nodes in the same set are adjacent.*

Theorem 4.10. *If $a+b$ is even then the Gaussian network generated by $\alpha = a+bi$ is a bipartite graph where the two sets of even nodes and odd nodes is the partition of nodes.*

Proof. We show that when $a+b$ is even the two disjoint sets for the bipartite graph G_α are the set of even nodes and the set of odd nodes defined above. That is, we show that when $a + b$ is even, odd nodes are adjacent only to even nodes and vice

versa. First consider the nodes that are endpoints of regular edges in the half-open square representation of G_α. Two nodes are adjacent if their addresses differ in one position by ± 1 or $\pm i$; That is, if a node is odd (even), then every neighbor is even (odd).

Next consider the boundary nodes. Let the node P_1 be a boundary node in G_α. Let the node P_2 be the neighbor of P_1 which is located outside of the basic square. By the above explanation, if P_1 is even (odd) then P_2 must be an odd (even) node. Let P_3 be the translation of the node P_2 to the basic square. Then, P_3 is obtained by subtracting $m\alpha$ from P_2 for $m \in \{1, -1, i, -i\}$. Since $a + b$ is even, P_3 is even (odd) provided that P_2 is even (odd). In other words, we have shown that if the boundary node P_1 is even (odd) its neighbor through the wraparound edge is odd (even). □

4.5 Degree-Three Pruned Gaussian Networks

This section introduces degree-three pruned Gaussian networks as an alternative to honeycomb rectangular tori.

Consider a Gaussian network G_α generated by $\alpha = a + bi$ where $0 < a \leq b$ and $a + b$ is even. The pruning technique used to construct a degree-three pruned Gaussian network works as follows: For each given node $x + yi$ in the network, the edge incident to node $(x + yi) + 1$ will be removed if and only if $x + yi$ is an odd node, where the addition is modulo α. Consequently, the edge incident to node $(x + yi) - 1$ will be removed if and only if $x + yi$ is an even node, where the subtraction is modulo α. Figure 4.3 shows the square representation of the pruned Gaussian network generated by $\alpha = 3 + 5i$.

Note that when if either $a = 0$ or $b = 0$, the degree-three pruned Gaussian network is a pruned torus of size $b \times b$ or $a \times a$. Thus, a degree-three pruned Gaussian network is a generalization of a honeycomb squared torus (pruned squared torus).

Theorem 4.12 shows that a degree-three pruned Gaussian network is regular of degree three provided that $a + b$ is even. First, let us define a regular graph.

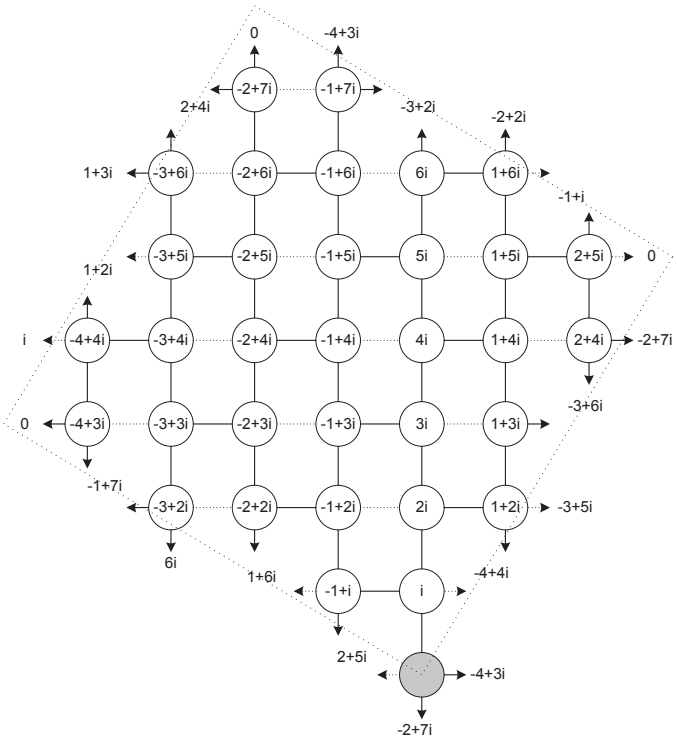


Figure 4.3: The square representation of the pruned Gaussian network generated by $\alpha = 3 + 5i$. The pruned edges are represented by dotted lines.

Definition 4.11. *A graph is said to be regular of degree r if and only if all the nodes of the graph have degree r.*

Theorem 4.12. *Let G_α be a Gaussian network generated by $\alpha = a + bi$ such that $0 < a \leq b$. Then the degree-three pruned Gaussian network defined by pruning G_α is regular of degree three if $a + b$ is even.*

Proof. By Theorem 4.10, G_α is a bipartite graph when $a + b$ is even. The two disjoint sets of the bipartite graph are the set of even nodes and the set of odd nodes. Each even (odd) node is adjacent to four odd (even) nodes. For each node, the pruning technique defined above removes one of its two horizontal edges as shown in Figure 4.4. Therefore, all nodes of the remaining network have exactly three neighbors and the resulting network is regular of degree three. □

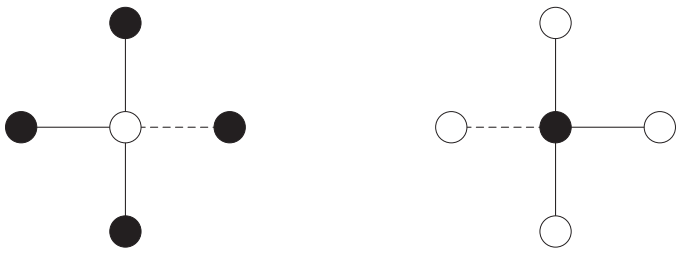


Figure 4.4: A node and its edges in a degree-three pruned Gaussian network. The dotted lines represent the pruned edges. Black circles represent even nodes and white circles represent odd nodes.

Next we prove that the resulting degree-three pruned Gaussian network is a Cayley graph, and thus it is node-symmetric.

4.5.1 Node Symmetry

Cayley graphs are known to be node-symmetric [35]. Therefore, in order to prove that a network is node-symmetric, it suffices to show that it is a Cayley graph.

Next we prove that the degree-three pruned Gaussian networks are Cayley graphs. In the proof we adapt the technique given in [34, 41].

Theorem 4.13. *Let G_α be a Gaussian network generated by $\alpha = a + bi$ such that $0 < a \leq b$. Then the degree-three pruned Gaussian network defined by pruning G_α is a Cayley graph.*

Proof. Let Γ be the set of Gaussian integers within the basic square $\mathcal{P}_\alpha$.

For each $(m + in), (u + iv) \in \Gamma$ define an operation $\oplus$ as follows (the addition on the left is the addition of complex numbers):

$$(m + in) \oplus (u + iv) = m + (-1)^{m+n}u + i(n + v) \mod \alpha$$

First, we show that Γ is a group under the operation $\oplus$ defined above.

- Closure: For any two points $(m + in), (u + iv) \in \Gamma$, the result of $(m + in) \oplus (u + iv)$ is calculated modulo α that is, the remainder upon division by α. Thus, the result is a point in Γ.

- Identity: 0 is the identity element under $\oplus$ because $\forall(m + in) \in \Gamma$ we have $(m + in) \oplus 0 = 0 \oplus (m + in) = (m + in)$.

- Associativity: $\forall(m+in), (u+iv), (z+it) \in \Gamma$ we have $((m+in) \oplus (u+iv)) \oplus (z + it) = (m + in) \oplus ((u + iv) \oplus (z + it))$. There are four cases to show this equality holds depending on whether $m + n$ and $u + v$ are even or odd. We prove the case when $m + n$ is even and $u + v$ is odd. For the other cases a similar proof can be given.

$$\begin{aligned}((m + in) \oplus (u + iv)) \oplus (z + it) &= (m + u + i(n + v)) \oplus (z + it) \\ &= m + u - z + i(n + v + t) \\ &= (m + in) \oplus (u - z + i(v + t)) \\ &= (m + in) \oplus ((u + iv) \oplus (z + it))\end{aligned}$$

- Inverse: For every $(m + in) \in \Gamma$ it can be checked that there is an inverse

such that

$$(m+in) \oplus (m+in)^{-1} = (m+in)^{-1} \oplus (m+in) = 0$$

where $(m+in)^{-1}$ is either $(-m-in)$ or $(m-in)$ depending on $m+n$ is even or odd.

The set

$$S = \{1, +i, -i\}$$

which is a subset of Γ (assuming the basic square is centered at 0) is a generator set for Γ because every element of Γ can be obtained by a combination of finitely many elements of S and their inverses under $\oplus$. Therefore, S is a generator set for Γ. Also, Γ has the same node set as the degree-three pruned Gaussian network generated by α. Every node $(x+iy) \in \Gamma$ is adjacent to the nodes $(x+iy) \oplus 1$, $(x+iy) \oplus i$, and $(x+iy) \oplus -i$. Thus, Γ is the degree-three pruned Gaussian network and is a Cayley graph. □

4.5.2 Diameter

In this section we use an approach analogous to that used in [41] to calculate the diameter of degree-three pruned Gaussian networks.

The diameter is a longest shortest path between any pair of nodes in a network. Since a degree-three pruned Gaussian network is node-symmetric, the diameter can be considered to be the longest shortest path from the origin. Consider the basic square defined by a Gaussian network. The square has four corners, but only one of them is included in the node set of the network. The other three corners of the basic square are actually multiples of α. In other words, they are modulo α translations of the origin. We assume the following conjecture that we have not proved.

Conjecture 4.14. *The distance of a node from the origin in a pruned Gaussian network is the minimum of the (pruned Gaussian) distance of this node from the*

four corners of the basic square.

We will use this fact to calculate the diameter of a degree-three pruned Gaussian network. But first let us find a lower bound for the diameter. It is shown in [36, Corollary 13] that the diameter of a Gaussian network generated by $\alpha = a + bi$, where $0 \leq a \leq b$ is b if $a^2 + b^2$ is even. Since in a degree-three pruned Gaussian network $a + b$ is always even, its diameter is at least b.

Theorem 4.15. *Let G_α be a degree-three pruned Gaussian network generated by $\alpha = a + bi$ such that $0 < a \leq b$. Assuming Conjecture 4.14, then the diameter d of the network is*

$$d = \begin{cases} b & \text{if } a \leq \lceil b/2 \rceil \\ 2k + r & \text{if } a > \lceil b/2 \rceil \text{ and } a + b = 3k + r \text{ for } r = 0, \pm 1. \end{cases}$$

Corollary 4.16. *The cost of the network defined by the product of the degree and the diameter is smaller in a degree-three pruned Gaussian network generated by $\alpha = a + bi$ ($0 < a < b$ and $a + b$ is even) compared to a Gaussian network with the same generator.*

Proof. Since the diameter of the Gaussian network generated by α is b the cost of the network is $c_1 = 4b$. The cost of the degree-three pruned Gaussian network generated by α is $c_2 = 3b$ if $a \leq \lceil b/2 \rceil$ and is $c_3 = 2a + 2b + r$ for $r = 0, \pm 1$ if $a > \lceil b/2 \rceil$. Since $2a < 2b$, we have $c_3 < c_1$. □

Note that if $a = b$ and $a + b = 3k + 1$ then the cost of a degree-three pruned Gaussian network is one more than that of a Gaussian network generated with the same generator.

We prove Theorem 4.15 by first proving a useful equation for finding the distance between the nodes.

Let $x_0 + y_0 i$ and $x_1 + y_1 i$ be the source node and the destination node, respectively. First we suppose $x_0 + y_0 i$ is an even node. Consider these two nodes in the

2D plane. Let $\Delta x = x_1 - x_0$ and $\Delta y = y_1 - y_0$. The distance between two nodes $x_0 + y_0 i$ and $x_1 + y_1 i$ is given by

$$\begin{cases} |\Delta x| + |\Delta y| & \text{if } |\Delta y| \geq |\Delta x| \\ 2\Delta x - s & \text{if } |\Delta y| < \Delta x \text{ and } \Delta x > 0 \\ 2|\Delta x| + s & \text{if } |\Delta y| < |\Delta x| \text{ and } \Delta x < 0 \end{cases} \tag{4.1}$$

where $s = 0(1)$ if $\Delta x + \Delta y$ is even (odd). If $x_0 + y_0 i$ is an odd node then the distance is given by

$$\begin{cases} |\Delta x| + |\Delta y| & \text{if } |\Delta y| \geq |\Delta x| \\ 2\Delta x + s & \text{if } |\Delta y| < \Delta x \text{ and } \Delta x > 0 \\ 2|\Delta x| - s & \text{if } |\Delta y| < |\Delta x| \text{ and } \Delta x < 0 \end{cases} \tag{4.2}$$

We now prove (4.1) and (4.2). In order to find the distance between the source and the destination we will trace a route similar to the dimension order routing algorithm. The route starts from the source to first correct Δx unless the required edge has been pruned. In that case, the route continues one step in dimension Y in the direction that corrects Δy. If Δy is already corrected then two extra edges must be added to the path: one edge adding an offset to Δy and one removing the added offset.

If $|\Delta y| \geq |\Delta x|$ no extra steps are required. Thus, the distance between source and destination is $|\Delta x| + |\Delta y|$. If $|\Delta y| < |\Delta x|$ then extra steps may be needed. Depending on the sign of Δx two different cases will be considered. We assume the source node is an even node. The argument for odd node is similar.

Case 1. $\Delta x > 0$. The route starts by alternating dimensions until $\Delta x - |\Delta y|$ edges in dimension X remain to be corrected. For each pair of remaining steps in dimension X, two additional steps are required to go back and forth in dimension Y. Note that, if there is only one step required to correct the X-dimension edge,

it can be done in one edge. Therefore, the route is completed in $2\lfloor\frac{\Delta x-|\Delta y|}{2}\rfloor$ more steps.

Case 2. $\Delta x < 0$. The route begin in dimension Y and continues to alternate dimensions as before, until $|\Delta x| - |\Delta y|$ edges in dimension X remain to be corrected. Then, if one step is required to correct the offset in dimension X, three additional steps are needed, for a total of $2\lceil\frac{|\Delta x|-|\Delta y|}{2}\rceil$ additional steps. In summary we can write the distance between $x_0 + y_0 i$ and $x_1 + y_1 i$ is

$$\begin{cases} |\Delta x| + |\Delta y| & \text{if } |\Delta y| \geq |\Delta x| \\ \Delta x + |\Delta y| + 2\lfloor\frac{\Delta x-|\Delta y|}{2}\rfloor & \text{if } |\Delta y| < \Delta x \text{ and } \Delta x > 0 \\ |\Delta x| + |\Delta y| + 2\lceil\frac{|\Delta x|-|\Delta y|}{2}\rceil & \text{if } |\Delta y| < |\Delta x| \text{ and } \Delta x < 0 \end{cases} \tag{4.3}$$

where $\Delta x = x_1 - x_0$ and $\Delta y = y_1 - y_0$ and $x_0 + y_0 i$ is an even node.

The term $2\lceil\frac{|\Delta x|-|\Delta y|}{2}\rceil$ equals $|\Delta x|-|\Delta y|$ if $\Delta x+\Delta y$ is even and $|\Delta x|-|\Delta y|+1$ if $\Delta x + \Delta y$ is odd. Similarly, $2\lfloor\frac{\Delta x-|\Delta y|}{2}\rfloor$ equals $\Delta x - |\Delta y|$ or $\Delta x - |\Delta y| - 1$ if $\Delta x + \Delta y$ is even or odd, respectively. Thus, the above equation can be simplified and be written as in Equation (4.1).

Now we are ready to prove Theorem 4.15. As stated earlier, when $a+b$ is even the diameter of the Gaussian network is b and therefore the diameter of the pruned Gaussian network is at least b.

Next we identify the situations in which the diameter is greater than b. As we prove this we will prove that the diameter equals b when $a \leq \lceil b/2 \rceil$. Consider the square representation of the Gaussian network G_α generated by $\alpha = a + bi$ $(0 < a \leq b)$ as shown in Figure 4.5. When $a = 0$ then the Gaussian network is a torus and that has been discussed thoroughly in [41]. In the following discussion we first assume b is even. Later we assume b is odd.

Let $M = x + yi$ be a node in G_α. Based on Conjecture 4.14 the distance of M from the origin is the minimum of the distances of M from the four corners of the basic square. Note that the term *distance* means using the edges of the pruned

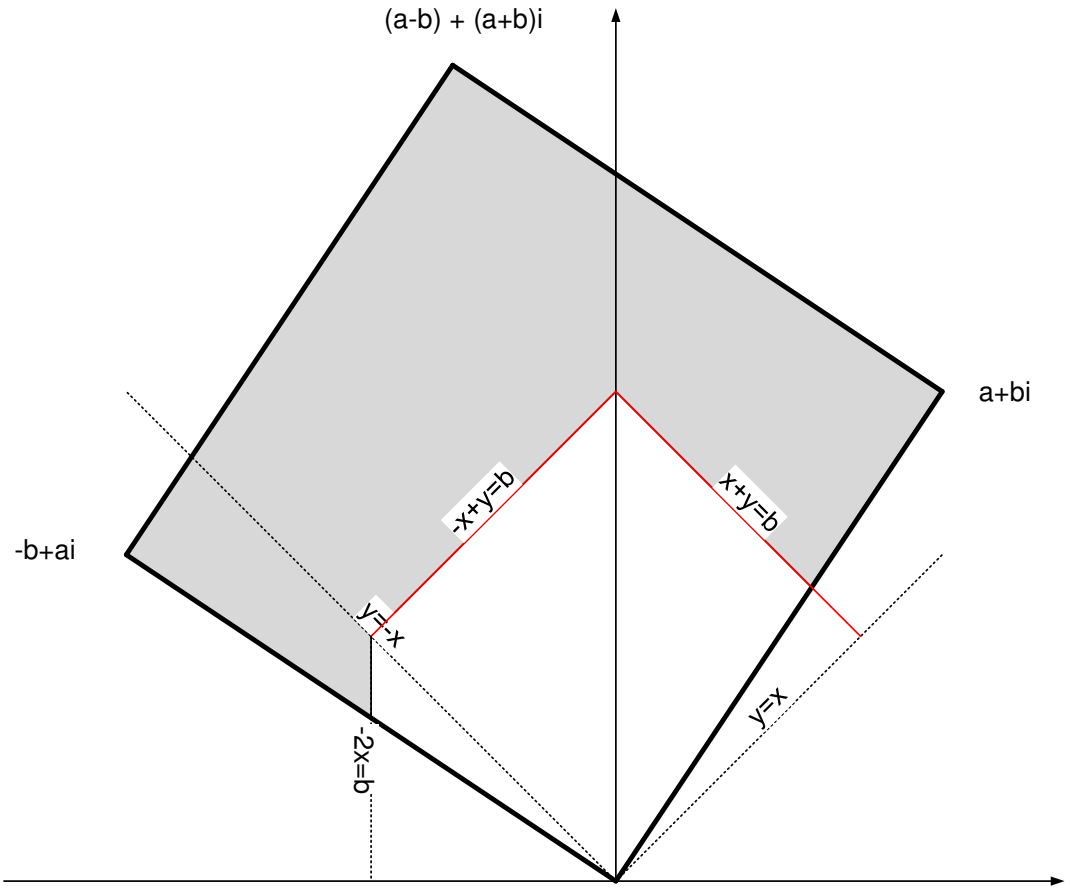


Figure 4.5: Shaded region shows the region where the distance from the origin might be greater than b.

graph. By Equation (4.1) the distance of M from the origin is $|x| + y$ if $y \geq |x|$; that is, M is in the region between the two lines $y = x$ and $y = -x$. The region with a distance greater than b from the origin and within these two lines lies within by $x + y > b$ and $-x + y > b$. If $y < |x|$ then by Equation (4.1) the region with a distance greater than b lies within $2|x| \geq b$. This region is shown in Figure 4.5.

Note that in the following discussion we consider the case where $a < b$. If $a = b$ then the line $y = \frac{b}{a}x$ which is the side of the square between the origin and $a + bi$ lies on the line $y = x$. In this case the following discussion will be simplified since some of the regions disappear. For the case $a < b$ the line $y = \frac{b}{a}x$ lies between the two lines $y = x$ and $x = 0$ as is shown in Figure 4.5 because the slope of this line is greater than 1 (since $0 < a < b$ we have $\frac{b}{a} > 1$). Using similar reasoning in later figures all the lines lie as shown.

Now, consider the corner which is the point $(a - b) + (a + b)i$. Assume the node $x + yi$ is in the basic square and let $\Delta x = x - (a - b)$ and $\Delta y = y - (a + b)$. If $|\Delta y| \geq |\Delta x|$ the region with a distance greater than b from $(a - b) + (a + b)i$ and the lines $y = x + 2b$ and $y = -x + 2a$ lies within the region $x + y < 2a - b$ and $-x + y < b$. If $|\Delta y| < |\Delta x|$ then the region with a distance greater than b lies within $2x \geq 2a - b$. The shaded regions in Figure 4.6 describe all the possible regions where distance of M from origin can be greater than b.

Considering the above constraints and looking at Figure 4.6 there are only two regions in which there could be a node with a distance greater than b from the two mentioned corners of the basic square. We call these two regions A and B. Region A is bounded by the two sides of the square originating from $a + bi$ and the lines $x + y = b$, $-x + y = b$, and $2x = 2a - b$. The region between the other two sides of the square originating from $-b + ai$ and bounded by the lines $x + y = 2a - b$, $-x + y = b$, and $-2x = b$ forms region B. These regions are shown in Figure 4.6.

Next, consider the remaining two corners of the basic square. Similar to the previous discussion we draw two pairs of lines, each pair of line begins in one of the two corners. These lines in addition to the previous lines and the basic square are shown in Figurer 4.7.

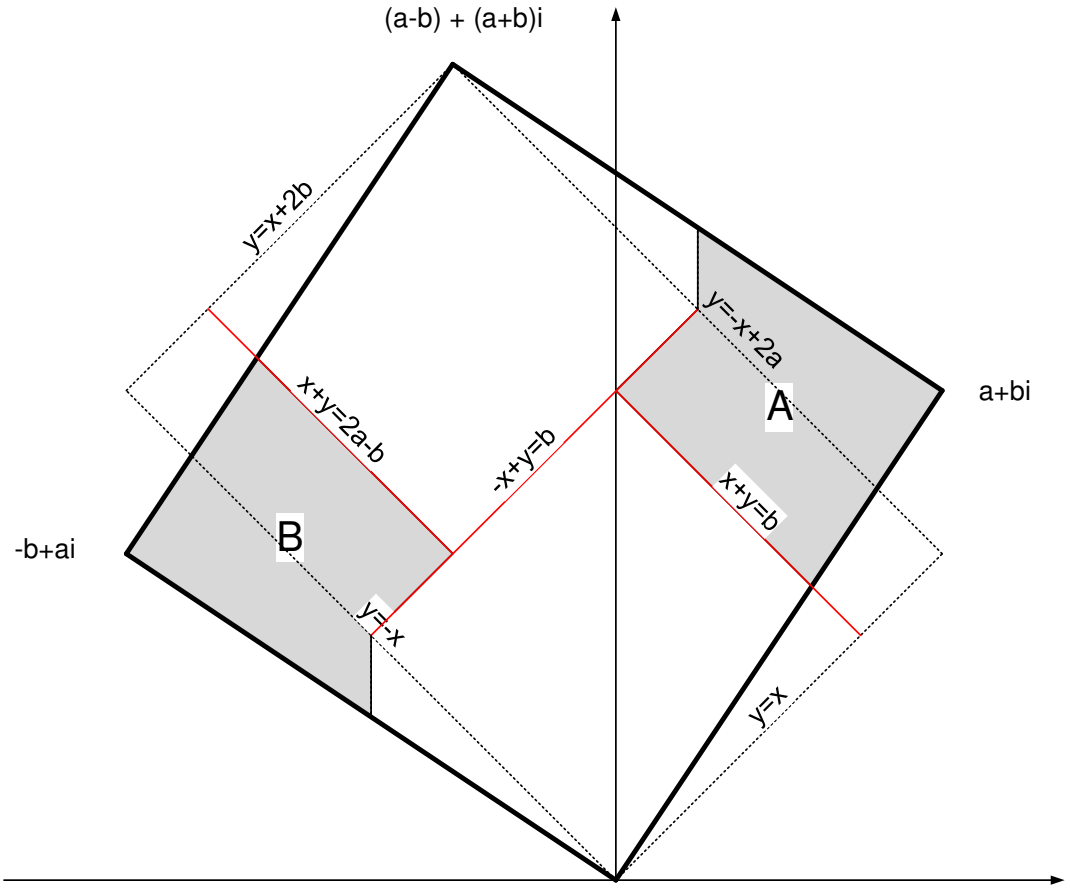


Figure 4.6: Shaded regions show the two regions where the distance from the origin and $(a-b)+(a+b)i$ might be greater than b.

Note that there is a hidden assumption in Figurer 4.7 (and so in all the following figures). The assumption is $2a - b > 0$. Let the node H be the intersection of the two lines $y = b$ and $y = -x + 2a$. Let $D = bi$. Since the length of the line DH is $2a - b$, when $2a - b \leq 0$ then the rectangle (which we call it A_1) region does not exist. In other words, if $a \leq \lfloor b/2 \rfloor$ then the diameter of the network is b.

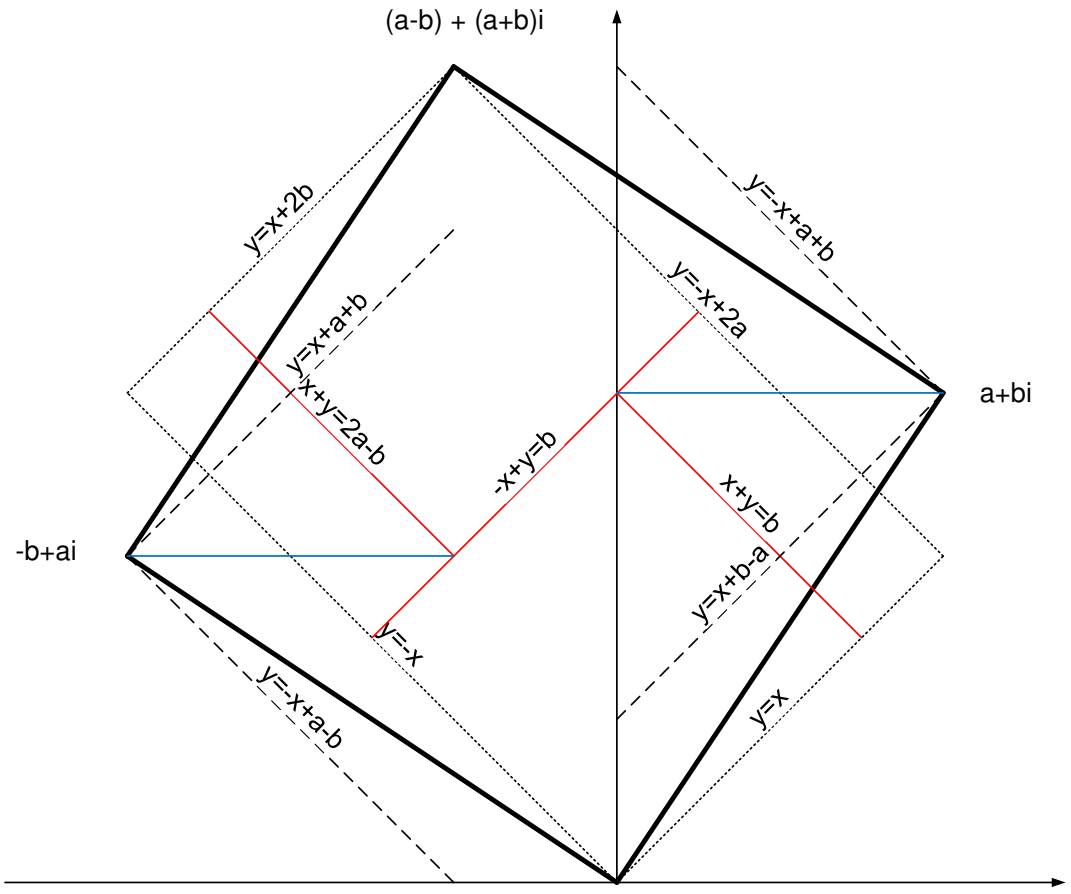


Figure 4.7: Basic square and the lines used to prove the diameter of the Gaussian network G_{a+bi} where $0 < a < b$.

First consider the region A. There cannot be any node where the distance is greater than b lying in the triangular shape bounded by the side of the square (the side between $a + bi$ and $(a - b) + (a + b)i$) and the two lines $y = -x + 2a$ and

$y = x + b - a$. Assume $P_1 = x_1 + y_1 i$ is such a node. By Equation(4.1), the distance of P_1 from $(a - b) + (a + b)i$ is $d_1 = 2(x_1 - (a - b)) - s_1$ where s_1 is 0 or 1 depending on whether P_1 is even or odd. By Equation(4.1), the distance of P_1 from $a + bi$ is $d_2 = 2(a - x_1) + s_1$. Since $d_1 + d_2 = 2b$ it is impossible to have $d_1 > b$ and $d_2 > b$.

There cannot be any node whose distance is greater than b lying in the triangle generated by the side of the square (the side between 0 and $a + bi$) and the two lines $y = x + b - a$ and $x + y = b$. On the contrary, assume $P_2 = x_2 + y_2 i$ is such a node. However, the nodes with the distance greater than b from $a + bi$ are contained in the region defined by $x + y < a$. In this case P_2 is not in this region.

Thus, the rest of nodes in A where distance is greater than b lie within the region bounded by $x + y = b$, $y = x + b - a$, $y = -x + 2a$, and $-x + y = b$. We call this rectangle A_1.

Next consider the region B. There cannot be any node where distance is greater than b lying in the triangular shape bounded by the side of the square (the side between $-b + ai$ and the origin) and the two lines $y = -x$ and $y = x + a + b$. Assume $P_3 = x_3 + y_3 i$ is such a point. By Equation(4.1), the distance of P_3 from $b + ai$ is $d_3 = 2(b - |x_3|) - s_3$ where s_3 is 0 or 1 depending on whether P_3 is even or odd. The distance of P_3 from the origin is $d_4 = 2(|x_3|) + s_3$. Since $d_3 + d_4 = 2b$ it is impossible to have $d_3 > b$ and $d_4 > b$.

There cannot be any node where distance is greater than b lying in the triangle generated by the side of the square (the side between $(a-b)+(a+b)i$ and $-b+ai$) and the two lines $y = x + a + b$ and $x + y = 2a - b$. Assume $P_4 = x_4 + y_4 i$ is such a node. However, the nodes where distance is greater than b from $-b+ai$ are contained in the region defined by $x + y > a$. P_4 is not in this region.

Thus, the rest of nodes in B where distance is greater than b lie within the region bounded by $x + y = 2a - b$, $y = x + a + b$, $y = -x$, and $-x + y = b$. We call this rectangle B_1.

The two regions A_1 and B_1 are shown in Figure 4.8.

Now let us take a closer look at regions A_1 and B_1. First consider region

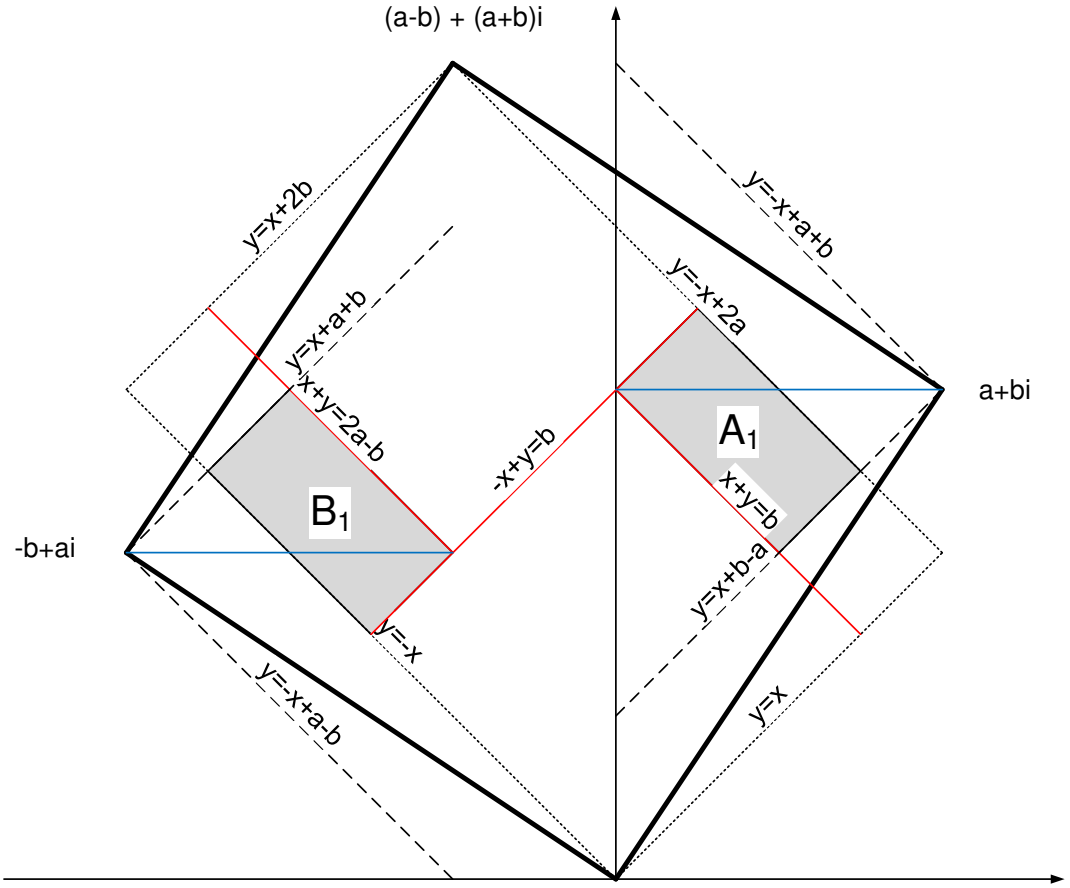


Figure 4.8: Shaded regions are the region A_1 and the region B_1 in which the network distance might be greater than b.

A_1 as shown in Figure 4.9. The coordinates of the nodes are as follows. $C = a - b/2 + (a + b/2)i$, $D = bi$, $E = a - b/2 + (3b/2 - a)i$, $F = a - b/2 + bi$, and $G = a + bi$. Note that since we are assuming for now that a and b are even integers, all five points have integer coordinates and so are nodes in the rectangle.

By (4.1) (the third case applies), the distance of any node P on the line CE from the corner G of the basic square is $2(a - (a - b/2)) + s = 2(b/2) + s = b + s$ where s is 0 or 1 . By (4.1) all the nodes that lie within the rectangle and between G and the line CE have distance less than b from G. Thus the only nodes where distance from 0 might be greater than b is the triangle CDE (excluding the points on the lines CD and DE, because the distance of any node on these lines from 0 is b). We call this triangle region A_2. This region is the shaded triangle shown in Figure 4.9. If a and b are both even, then C and E have integer coordinates. Thus, the line CE is also included in the region A_2.

If a and b are both odd, then C and E are not nodes in the network. Instead the nodes in the network lie on the two lines $x = a-(b+1)/2$ and $x = a-(b-1)/2$. These two lines are the L_1 and L_2 lines shown in Figure 4.10, respectively. By (4.1), the distance of any node on the line L_2 from G is at most b. Thus this line is not included in the region A_3. But the line L_1 is included in the region A_3.

Now consider region B_1 as shown in Figure 4.11. The coordinates of the nodes are as follows. $C' = -b/2 + b/2i$, $D' = a - b + ai$, $E' = -b/2 + bi$, $F' = -b/2 + ai$, $F'' = -b/2 + 1 + ai$, and $G' = -b + ai$. Note that since we are assuming for now that a and b are even integers, all five points have integer coordinates and so are nodes in the rectangle. By (4.1) (the second case applies), the distance of any node on the line $C'E'$ from the corner of the basic square $G' = -b + ai$ is $2(b - b/2) - s = 2(b/2) - s = b - s$ where s is 0 or 1. By (4.1) all the nodes that lie within the rectangle and between the point G' and the line $C'E'$ and including the points on the line $C'E'$ have distance less than or equal to b from G'. Thus the only region where the distance might be greater than b is the shaded triangle shown in Figure 4.11 (excluding the points on the lines $C'D'$ and $D'E'$). Note that $F'' = F' + 1$. We call this triangular region B_2. If a and b are both even, then C'

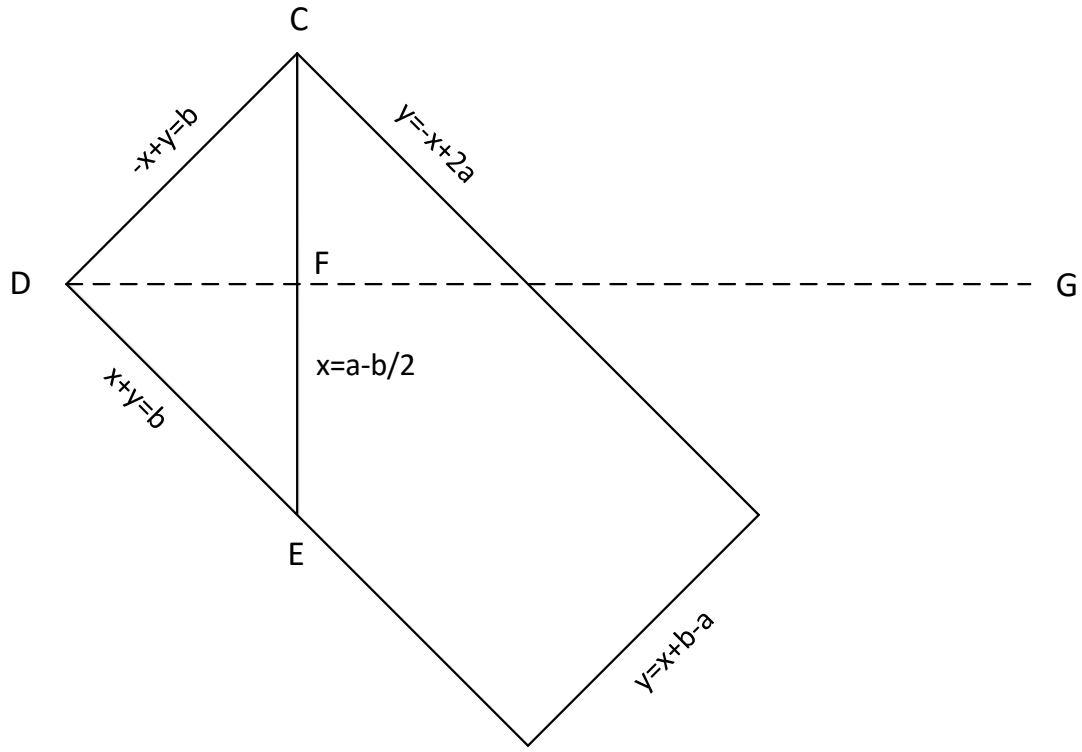


Figure 4.9: The rectangle is the region A_1 and the shaded triangle is the region A_2 in which the network distance might be greater than b for even b.

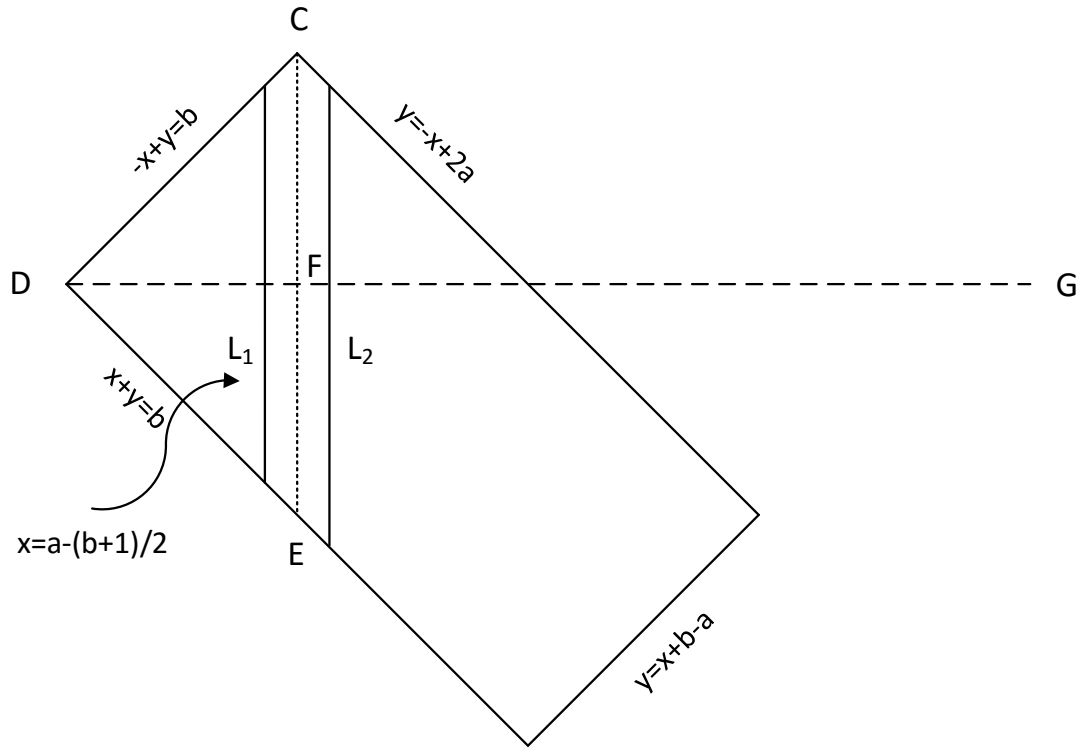


Figure 4.10: The rectangle is the region A_1 and the shaded triangle is the region A_3 in which the network distance might be greater than b for odd b.

and E' as well as F' and F'' have integer coordinates. Thus, the line $x = F''$ is also included in the region B_2.

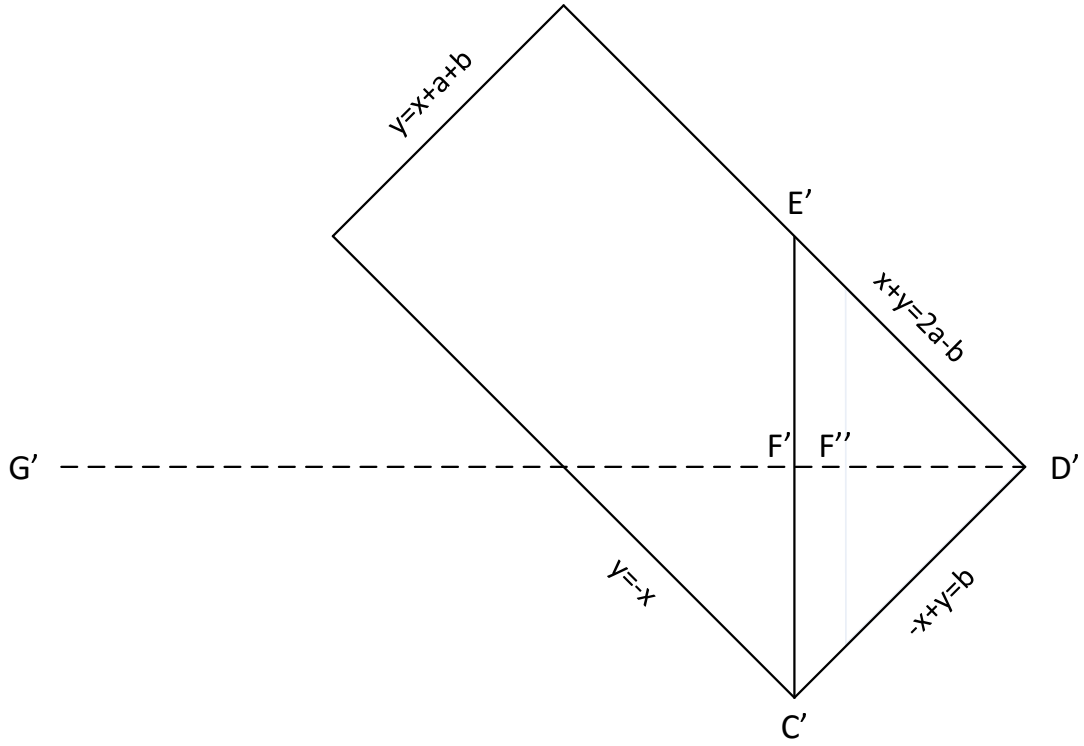


Figure 4.11: The rectangle is the region B_1 and the shaded triangle is region B_2 in which the network distance might be greater than b for even b.

If a and b are both odd, then C' and E' are not nodes in the network. Instead the nodes in the network lie on the two lines $x = -(b-1)/2$ and $x = -(b+1)/2$. These two lines are the L_1 and L_2 lines shown in Figure 4.12, respectively. By (4.1), the distance of any node on the line L_2 from G' is at most $b-1$. Thus this line is not included in region B_3. But the line L_1 is included in the region B_3.

Simple math shows that the triangle CDE and $C'D'E'$ in the previous figures have the same size. All triangles are isosceles triangle and the size of the sides CE or $C'D'$ is $2a - b$. Comparing Figure 4.9 and Figure 4.11, region A_2 is larger than region B_2 when a and b are both even. Although Figure 4.10 and Figure 4.12 show

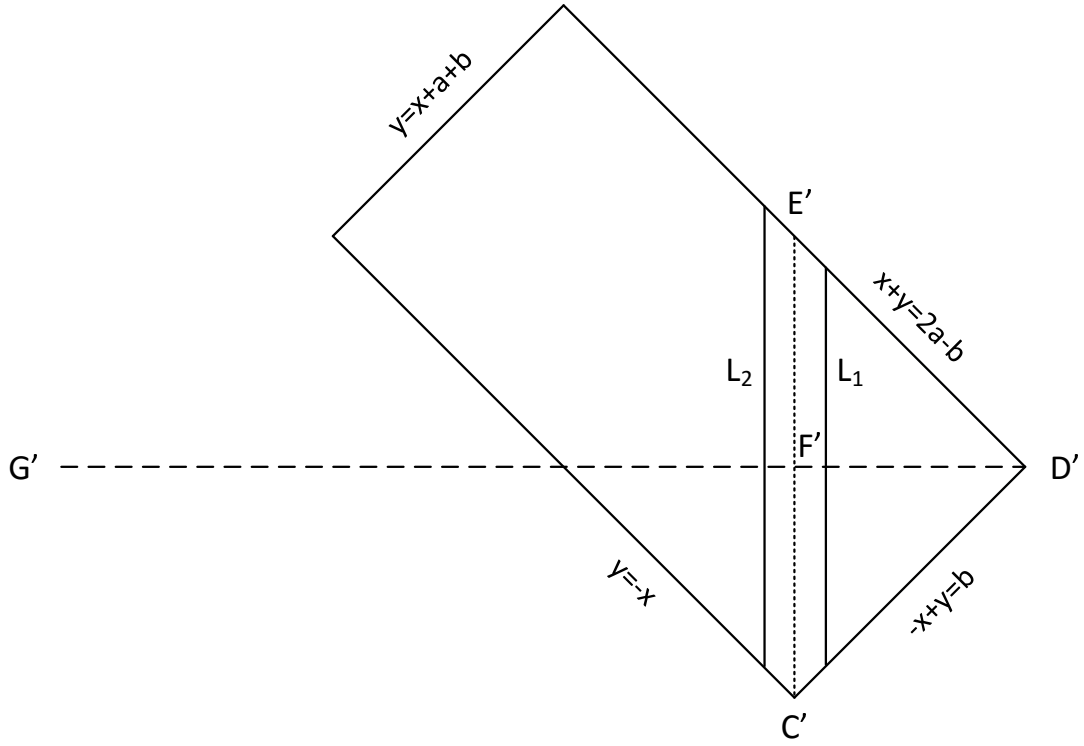


Figure 4.12: The rectangle is the region B_1 and the shaded triangle is region B_3 in which the network distance might be greater than b for odd b.

that region A_2 and region B_2 have the same size when a and b are both odd region A_2 may have some nodes with a greater distance from the corner. Consider the nodes on the line L_1 in the figures. By (4.1), the distance of any node on this line from G is at most $b+2$ in Figure 4.10. However, the distance of any node on the line L_1 in Figure 4.12 is at most $b+1$.

Next we find the value for the *diameter nodes*. A Diameter node is a node whose distance from 0 is the diameter of the network. They must lie in the regions A_2 or B_2 for even b (A_3 or B_3 for odd b). By (4.1) the distance of any node in region A_2 from $-b+ai$ is at least $2b$. This is because the X offset is $2b$. However, the maximum distance of any node in region A_2 from the other three corners is at most $2a$. (By (4.1), the distances of C from 0, E from $(a-b)+(a+b)i$, and D from $a+bi$ are $2a$.) Since $a < b$, to find diameter nodes in region A_2 (A_3) the corner $-b+ai$ is not considered. Similarly, to find diameter nodes in region B_2 (B_3) the corner $a+bi$ is not considered.

Lemma 4.17. *Let G_α be a degree-three pruned Gaussian network generated by $\alpha = a+bi$ such that $0 < a \leq b$. Let $u+(b \pm v)i$ specify the nodes in the region A_2 where u and v are integers such that $0 < u \leq \lfloor b/2 \rfloor$ and $0 \leq v \leq u-1$. Then any node with $v \geq 2$ cannot be a diameter node.*

Proof. Without loss of generality assume the node $P_1 = u+(b-v)i$ for $v \geq 2$ is a diameter node. If v is an even number, then by (4.1) the node $P_2 = u+bi$ and P_1 have the same distance from $a+bi$. Now consider the node $P_3 = u-1+bi$. This node is within the region A_2 since $v \geq 2$ implies $u \geq 3$. The distance of P_3 from the three corners of the basic square (all except for $-b+ai$) is greater than the distance of P_1 from the three corners. Thus, P_1 is not the node with the longest distance from the origin in the network.

Now, if v is an odd number. Then by (4.1) the node $P_4 = u+(b-1)i$ and P_1 have the same distance from $a+bi$. Now consider the node $P_5 = u-1+(b-1)i$. (Again this node is within the region A_2.) The distance of P_5 from the two corners $a+bi$ and 0 is greater than that of P_1. Thus, P_1 is not the node with the longest distance from the origin in the network. □

Lemma 4.18. *Let G_α be a degree-three pruned Gaussian network generated by $\alpha = a + bi$ such that $0 < a \leq b$. Let $a - b - u + (a \pm v)i$ specifies the nodes in the region B_2 where u and v are integers such that $0 < u < \lfloor b/2 \rfloor$ and $0 \leq v \leq u - 1$. Then any node with $v \geq 2$ cannot be a diameter node.*

Proof. Without loss of generality assume the node $P_1 = a - b - u + (a - v)i$ for $v \geq 2$ be the diameter. If v is an even number, then by (4.1) the node $P_2 = a - b - u + ai$ and P_1 have the same distance from $-b + ai$. Now consider the node $P_3 = a - b - u + 1 + bi$. This node is within the region B_2 since $v \geq 2$ implies $u \geq 3$. The distance of P_3 from the three corners of the basic square (all except for $a + bi$) is greater than that of P_1. Thus, P_1 is not the node with the longest distance from the origin in the network.

Now, if v is an odd number, then by (4.1) the node $P_4 = a - b - u + (a - 1)i$ and P_1 have the same distance from $-b + ai$. Now consider the node $P_5 = a - b - u + 1 + (a - 1)i$. (Again this node is within the region B_2.) The distance of P_5 from the two corners $-b + ai$ and 0 is greater than that of P_1. Thus, P_1 is not the node with the longest distance from the origin in the network. □

The following theorem is a direct result of Lemma 4.17 and Lemma 4.18.

Theorem 4.19. *Let G_α be a degree-three pruned Gaussian network generated by $\alpha = a + bi$ such that $0 < a \leq b$. Let $u + (b \pm v)i$ and $a - b - u + (a \pm v)i$ be the nodes in the region A_2 and the region B_2, respectively. (u and v are integers such that $0 < u \leq \lfloor b/2 \rfloor$ and $0 \leq v \leq u - 1$.) Then any diameter node has either $v = 0$ or $v = 1$.*

Now we are ready to find the value for diameter nodes. Consider the region A_2. By Theorem 4.19 first assume the node $P_1 = u + (b - 1)i$ for $0 < u \leq \lfloor b/2 \rfloor$ is a node with the longest distance from 0. P_1 must be an odd node. If P_1 is an even node then consider the node $P_2 = u + bi$ which is an odd node. By (4.1) the distance of P_2 from $a + bi$ is greater than the distance of P_1 from $a + bi$. In addition, the distance of P_2 from 0 is greater than the distance of P_1 from 0. Thus,

P_1 is not the node with the longest distance in the network. This means P_1 must be an odd node.

The distance of P_1 from 0 is $d_1 = u + b - 1$. The distance of P_1 from $a + bi$ is $d_2 = 2(a-u)+1$. We do not consider the distance of P_1 from $a-b+(a+b)i$ since this distance is greater than d_1. Since P_1 is not a diameter node then $d_2 \geq d_1$. If $d_2 \neq d_1$ there exists a node P_1' whose distance from $a+bi$ is d_1. Since d_1 is odd, P_1' has to be an odd node. The closest odd node to P_1 which is also closer to $a + bi$ is the node $u + 1 + bi$. Let P_1' be that node. Now consider the node $P_1'' = u + bi$. The distance of this node from $a + bi$ and also from 0 and from $(a - b) + (a + b)i$ is greater than d_1. Thus, d_1 cannot be the diameter.

The only possible case is $d_1 = d_2$. If we solve for $d_1 = d_2$ we have

$$\begin{aligned} u + b - 1 &= 2(a - u) + 1 \\ b - 2a &= -3u + 2 \\ b - 2a + 3a - 3a &= -3u + 3 - 1 \\ a + b &= 3(a - u + 1) - 1; \end{aligned}$$

that is, $a + b = 3k - 1$ for $k = a - u + 1$. In this case the diameter of the network is

$$u + b - 1 = a - k + 1 + b - 1 = 2k - 1.$$

Next consider the node $P_3 = u + (b + 1)i$ which is also a diameter node for $u = a - k + 1$ because the distance of P_3 from $a - b + (a + b)i$ and from $a + bi$ is also d_1. There is one more node with the same distance from $a - b + (a + b)i$ and 0. This node is $P_4 = u - 1 + bi$ (note that the distance of P_4 from $a + bi$ is greater than d_1).

Now consider the region B_2. Following the same argument and considering Theorem 4.19 and assuming the node $P_5 = a - b - u + (a - 1)i$ is a diameter node

we have

$$\begin{aligned} |a-b|+u+a-1 &= 2(a-u) \\ a+b &= 3(a-u+1)-2; \end{aligned}$$

that is, $a+b=3k-2$ for the same k obtained before. But this is not true because in the above discussion we had $a+b=3k-1$. This means the assumption P_5 is a diameter node is wrong. However, if we assume $P_6=a-b-u+ai$ be the diameter then we have

$$\begin{aligned} |a-b|+u+a &= 2(a-u) \\ a+b &= 3k-1, \end{aligned}$$

which is the same as the equation obtained when we were considering the region A_2. In short, we have shown that when $a+b=3k-1$ the diameter of the network is $2k-1$. In this case, there are three nodes in the region A_2 and one node in the region B_2 which have this distance from the origin.

Next, consider the region A_2 again. This time assume the node $P_7=u+bi$ is a diameter node. Since P_7 lies on the line $y=b$ it has the same distance $d_7=u+b$ from 0 and from $a-b+(a+b)i$. In this case P_7 can be either an odd or an even node.

Case 1. If P_7 is an even node, then by (4.1) the distance of P_7 from $a+bi$ is $d_8=2(a-u)$. Similar to the argument given above, d_7 has to be equal to d_8. If we solve for $d_7=d_8$ we have

$$\begin{aligned} u+b &= 2(a-u) \\ a+b &= 3(a-u); \end{aligned}$$

that is, $a+b=3k$ for $k=a-u$. In this case the diameter of the network is

$$u+b=a-k+b=2k$$

If we consider the region B_2 we also get the same result for the node $P_8 = a - b - u + ai$.

Case 2. If P_7 is an odd node. then by (4.1) the distance of P_7 from $a + bi$ is $d_9 = 2(a - u) + 1$. Again d_7 and d_9 must be the same. If we solve for $d_7 = d_9$ we have

$$u + b = 2(a - u) + 1$$
$$a + b = 3(a - u) + 1;$$

that is, $a + b = 3k + 1$ for $k = a - u$. In this case the diameter of the network is

$$u + b = a - k + b = 2k + 1.$$

It is easily verifiable that no node in the region B_2 has such a distance from the origin. This is the case where the diameter node only lies within the region A_2.

Note that in our argument to find the diameter we obtained the diameter by finding the longest distance from node 0 which is an even node. By Equation (4.2) in some situations the distance from an odd node is different than the distance from an even node. However, by node symmetry we will obtain the same result even if we find the diameter by calculating the distances from an odd node.

4.5.2.1 Comparison with Torus

The diameter of the honeycomb torus is given only for a special case where it has an square exterior shape (HST_m or $\text{HReT}(m, m/2)$). A HST_m has $n = m^2$ nodes and its diameter is m [41, 44]. When m is a multiple of two there is a pruned Gaussian network generated by $m/2 + mi$. This network also has diameter m but with $5m^2/4$ nodes. Therefore, a degree-three pruned Gaussian network provide 25% more nodes than the honeycomb torus of the same diameter.

In general, the diameter of a degree-three pruned Gaussian network is always less than the diameter of the pruned torus network of the same size. Consider a

degree-three pruned Gaussian network generated by $\alpha = a + bi$. We just show the case when $a \leq b/2$ in the last paragraph.

Let $a > \frac{1}{2}b$. Or, let $a = xb$ for $x > 1/2$. Then the number of nodes in the pruned Gaussian network is $N = a^2 + b^2 = x^2b^2 + b^2 = b^2(1 + x^2)$. The diameter of this pruned Gaussian network is $d_1 \approx \frac{2a+2b}{3} = \frac{2xb+2b}{3} = \frac{2}{3}b(1+x)$. The maximum value for the diameter of a pruned torus with N number of nodes is $d_2 = \sqrt{N} = b\sqrt{1+x^2}$.

Next, we show that d_2 is always greater than d_1. That is: $b\sqrt{1+x^2} > \frac{2}{3}b(1+x)$ or $\sqrt{1+x^2} > \frac{2}{3}(1+x)$.

Consider the inequality $5x^2-8x+5 > 0$. The minimum of $5x^2-8x+5$ occurs at $x = 8/10 = 4/5$. (This can be obtained by differentiating the equation and finding the root i.e. $10x - 8 = 0$.) This is minimum because the second differentiation is positive, i.e. the second differentiation of $10x-8$ is 10). Thus, the minimum value of this equation is $5(4/5)^2 - 8(4/5) + 5 = 9/5$. Thus, the inequality is always true.

Now we have:

$$\begin{aligned} 5x^2 - 8x + 5 &> 0 \\ 9(1 + x^2) &> 4(1 + x^2 + 2x) \\ 1 + x^2 &> \frac{4}{9}(1 + x)^2 \\ \sqrt{1 + x^2} &> \frac{2}{3}(1 + x) \end{aligned}$$

Thus, we showed that d_2 is always greater than d_1.

4.5.3 Routing

This section presents a shortest path routing algorithm for degree-three pruned Gaussian networks. Basically, the routing algorithm is similar to the dimension-order routing algorithm.

The route starts from the source to first correct the offset in dimension X one unit unless the required edge has been pruned. In that case, the route continues

one step in dimension Y in the direction that corrects the offset in dimension Y. This process continues by iteratively moving one step along each one of the two dimensions. When the offset in dimension X is corrected, the route continues to correct any remaining offset in dimension Y.

Since a degree-three pruned Gaussian network is node-symmetric, a shortest path routing from S to D can be found as follows. First, find the route from 0 to $D - S$ and then add S to each node in this route, where all operations are done modulo α. In the process of finding the route from 0 to $D - S$ we need to find distances.

Assuming Conjecture 4.14, the shortest path from 0 to $D - S$ is found by calculating the minimal distance from $D - S$ to the four corners of the basic square using Equation (4.1) if S is even. If S is odd then the situation is different because S and 0 do not have the same edge set. In this case we can shift the basic square (and also S and D) one unit to the right so that each corner is an odd node. Then we use Equation (4.2) to find the distances.

Algorithm 4.20 presents the routing algorithm for degree-three pruned Gaussian networks.

Algorithm 4.20. *Shortest path routing for degree-three pruned Gaussian network G_α where $\alpha = a + bi\,(0 \leq a \leq b$ and $a, b \in \mathbb{Z})$ is the network generator.*

Input: *The source node $S = x_0 + y_0 i$ and the destination node $D = x_1 + y_1 i$ where $x_j, y_j \in \mathbb{Z}$.*

Output: *A shortest path from S to D.*

1: *Calculate $D - S \mod \alpha$ and write it as $\Delta x + \Delta y i$.*

2: *Assume Conjecture 4.14. Among the four corners of the basic square choose the one with the minimum distance from $\Delta x + \Delta y i$ using Equation (4.1) if S is even. If S is odd use Equation (4.2). Write the minimum distance as $\Delta x' + \Delta y' i$.*

3: ***while*** $\Delta x' \neq 0$ ***or*** $\Delta y' \neq 0$ ***do***

4: ***if*** $\Delta x' \neq 0$ ***then***

5: ***if*** $\Delta x' > 0$ ($\Delta x' < 0$) ***and*** *the current node is odd (even)* ***then***
6: *Go to line 11*
7: ***else***
8: *Go one step along dimension X in the direction to correct* $\Delta x'$ *and decrease* $\Delta x'$
9: ***end if***
10: ***else***
11: *Go one step along dimension Y in the direction to correct* $\Delta y'$ *and decrease* $\Delta y'$. *If* $\Delta y' = 0$ *then go one step up or down.*
12: ***end if***
13: ***end while***

In line 1 of Algorithm 4.20, $D - S \mod \alpha$ needs to be calculated. It is explained in Section 4.4 that only one corner of the basic square is included in the Gaussian network. This node (the bottom right node) which we refer to it as C_{u+vi} is in the form $u\alpha + vi\alpha = (u + vi)\alpha$ for some $u, v \in \mathbb{Z}$. A given Gaussian integer w is located in the tile with corner C_{u+vi} if and only if $w - \gamma\alpha \in \mathcal{P}_\alpha$ for $\gamma = u + vi$ [24]. It is shown in [24] that for every Gaussian integer $w = x + yi$ there exist unique $r, s \in \mathbb{R}$ such that $w = r\alpha + si\alpha$. It is easily verifiable that r and s can be found by solving the following matrix equation: (The determinant of the 2×2 matrix is nonzero so we can solve the equation.)

$$\begin{bmatrix} a & -b \\ b & a \end{bmatrix} \begin{bmatrix} r \\ s \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} \tag{4.4}$$

Solving the above equation, we have $u \leq r < u + 1$ and $v \leq s < v + 1$ for some $u, v \in \mathbb{Z}$. Then we have

$$w \mod \alpha = w - (u + vi)\alpha \tag{4.5}$$

The routing algorithm is illustrated in the following example.

Example 4.21. *Let* $\alpha = 6 + 8i$. *Algorithm 4.20 finds a shortest path from* $A =$

$-2+11i$ to $B=3+5i$ in degree-three pruned G_α. First, $C=B-A=5-6i$ is calculated. Using Equation (4.4) we have $r=-18/100$ and $s=-76/100$. Since $-1 \le r < 0$ and $-1 \le s < 0$ the tile containing C has the corner labeled by $(-1-i)\alpha$. Using Equation (4.5), $C \mod \alpha = 5-6i-(-1-i)\alpha = 3+8i$. Next, we find the minimal distance from $3+8i$ to the four corners of the basic square using Equation (4.2) (because the source node A is an odd node), Thus, the corner $6+8i$ is the closest corner to $3+8i$ with distance 5. We write this distance as a complex number $\Delta x' + \Delta y' i = -3$. Finally, following the lines 3 to 13 of Algorithm 4.20 the route from A to B of length 5 is as follows: $A = -2+11i, -3+11i, -3+10i, -4+10i, -4+11i, -5+11i \equiv 3+5i = B$. This route is shown in Figure 4.13.

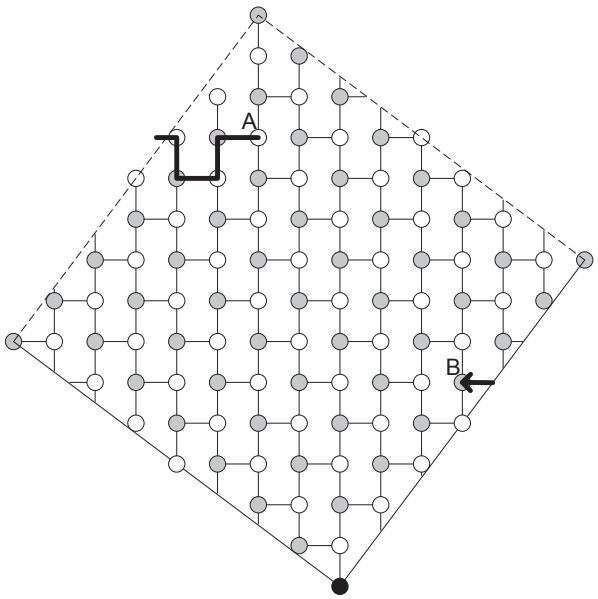


Figure 4.13: Shortest path route from $A=-2+11i$ to $B=3+5i$ in G_{6+8i} using Algorithm 4.20.

4.5.4 Broadcasting

In this section we provide a broadcasting algorithm for degree-three pruned Gaussian networks. We adopt the same approach provided in [43] to construct the broadcasting algorithm for degree-three pruned Gaussian networks. We assume the all-port model; that is, at every step of the algorithm a node can send a broadcast message to all of its neighbors.

4.5.4.1 Broadcasting in a Part of a Square

Consider the set of nodes in the 2D degree-three grid which are contained in a square whose diagonals are $2b$ (for some integer $b > 0$) and the slope of its sides are ± 1. First, let b be even. Let the square be centered at the origin. A *ball of radius* b and center at 0 is the set of nodes of distance less than or equal to b from 0 where the distance is calculated by (4.1). This ball is bounded by the square and it is the shaded region in Figure 4.14.

Consider the two lines $y = x$ and $y = -x$ in Figure 4.14. These lines divide the square into four geographic regions: up, down, left, and right. In the up and down regions for a given node $P = x + yi$ we have $|y| \geq |x|$. Thus, the distance of P from 0 is $|x| + |y|$ which is less than or equal to b in the square.

Consider the line $x = b/2$ in the right region. If $P = x + yi$ is located in the right region by (4.1) its distance from 0 is at most $2b$ and this occurs when P lies on the line $x = b/2$. Now consider the line $x = -b/2$. If $P = x + yi$ is located in the left region by (4.1) its distance from 0 is at most $2b + 1$ and this occurs when P lies on the line $x = -b/2$ for odd P. If P is even and lies on the line $x = -b/2$ then by (4.1) its distance from 0 is $2b$. Thus, only the even nodes on this line are included in the ball of radius b.

If b is odd, then place the square in a way that its center is an odd node. Note that we consider the center node to be origin or node 0, at the same time we say this node is odd. Therefore, from now on whenever b is odd our definition of odd nodes and even nodes are exchanged. That is, a node $x + yi$ is odd (even) when

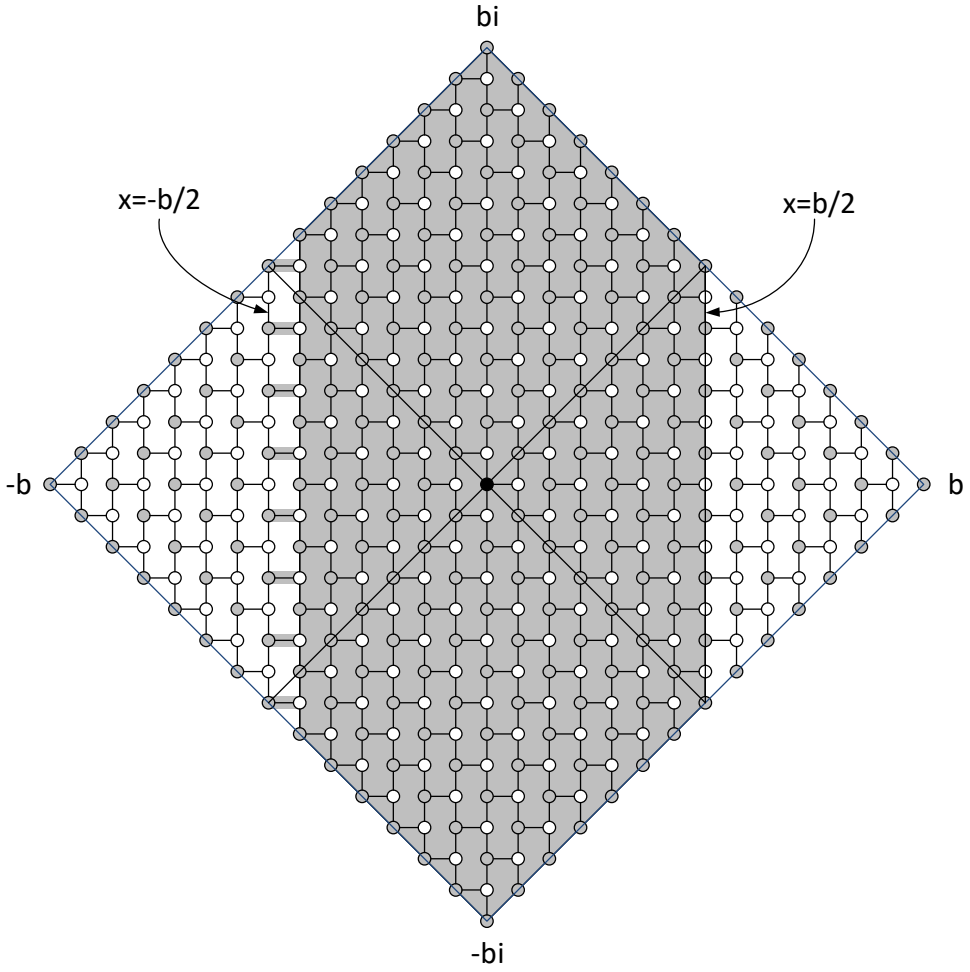


Figure 4.14: A ball of radius $b = 14$ in the square with diagonal equal to $2b$.

$x + y$ is even (odd). This simplifies and unifies our discussion.

Assume b is odd. Then the ball of radius b is defined by calculating the distances using (4.2) and is very similar as before except the two lines $x = -b/2$ and $x = b/2$ are replaced with the two lines $x = -(b+1)/2$ and $x = (b-1)/2$, respectively. Again, the odd nodes on the line $x = -(b+1)/2$ are not included in the ball.

The reason we choose the square in this section is that later we show that a Gaussian network G_α generated by $\alpha = a + bi$ can be placed within this square in a way that each corner of G_α lies on a side of the square.

Suppose the center of the square is the node with the broadcast message. We show a very simple broadcasting algorithm that broadcasts the message in the shaded region shown in Figure 4.14 in at most b steps.

First, a skeleton tree similar to the one given in [43] is constructed as follows for the case when b is even. Start from node 0 (center of the square) and generate four paths as follows. Path P_1 is built by iteratively moving right and up. P_2 is built by iteratively moving up and left. P_3 is built by iteratively moving down and left and P_4 is built by iteratively moving right and down. These paths are continued until reaching the borders of the square. The length of each of these paths is b. If b is odd, then the order of the moves will be changed; that is, path P_1 is built by iteratively moving up and right (instead of right and up), P_2 is built by iteratively moving left and up, P_3 is built by iteratively moving left and down and P_4 is built by iteratively moving down and right. Figure 4.15 shows the skeleton tree for a square with $b = 14$.

The skeleton tree defined above divides the square into four geographic regions: *up*, *left*, *down*, and *right*. Next we describe how to extend the skeleton tree into a spanning tree. Once we have a spanning tree constructed, the broadcast message can be propagated from the root (node 0) to all the nodes in the tree.

Consider the up region. Start from the even nodes on the path P_1 (apart from node 0) and the odd nodes on the path P_2 and construct paths by going up along direction Y until reaching the borders of the square. Similarly, in the down region start from the even nodes on the path P_4 (apart from node 0) and the odd nodes

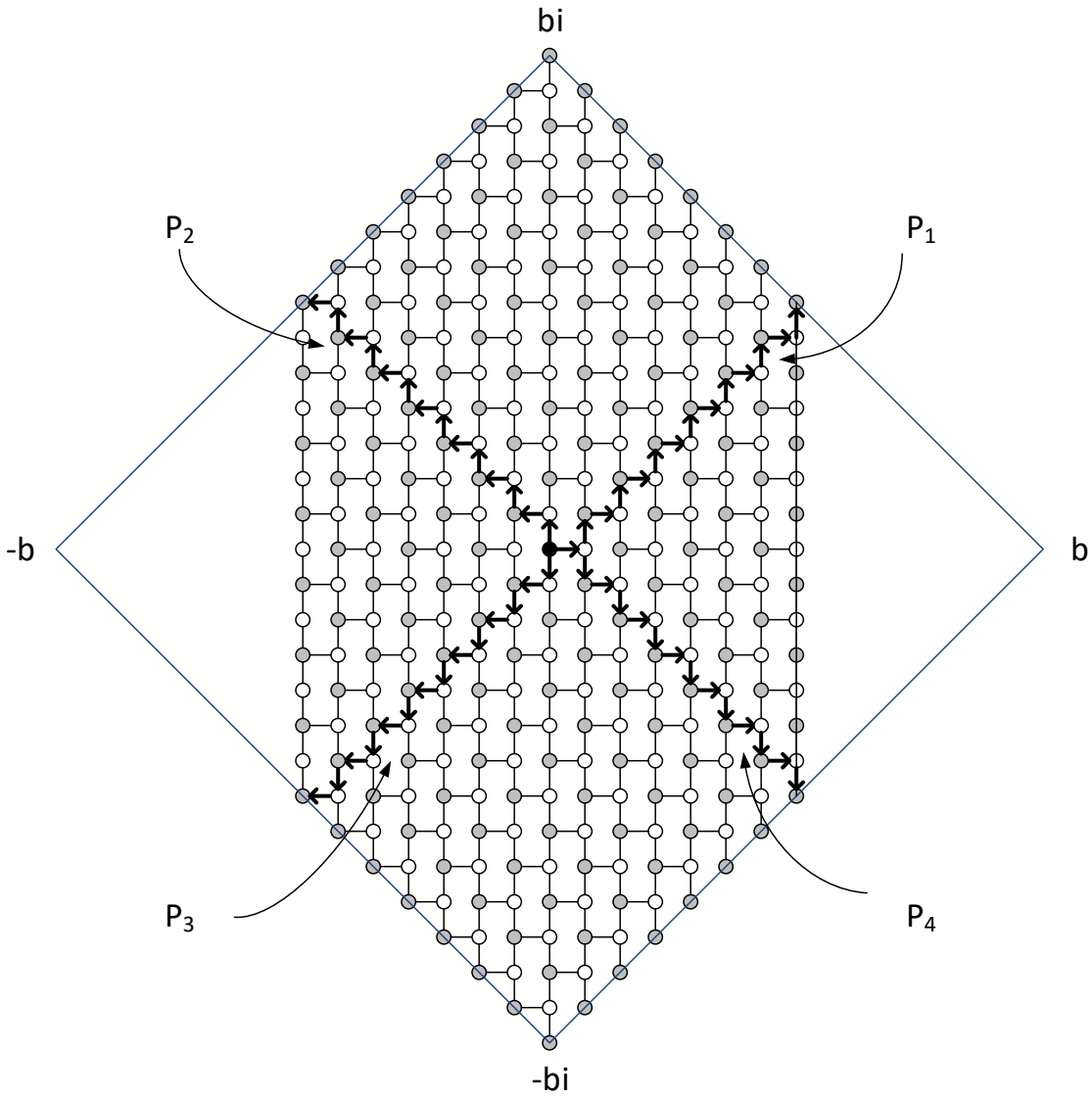


Figure 4.15: The skeleton tree for the broadcasting algorithm.

on the path P_3 and construct paths by going down along direction Y until reaching the borders of the square.

Consider the right region. The path P_1 is extended as follows. For every odd node $x + yi$ in the path if x is even and $x > 0$ construct a path from $x + yi$ by iteratively moving one step down, right, up, and right until reaching the line $x = b/2$. The path P_4 is extended as follows. For every odd node $x + yi$ in the path if x is odd and $x > 1$ construct a path from $x + yi$ by iteratively moving one step up, right, down, and right until reaching the line $x = b/2$.

Now consider the left region. The path P_2 is extended as follows. For every even node $x+yi$ in the path if x is odd construct a path from $x+yi$ by iteratively moving one step down, left, up, and left until reaching the line $x = -b/2$ ($x = -b/2+1$) if $x+yi$ is even (odd). The path P_3 is extended as follows. For every even node $x+yi$ in the path if x is even and $x < 0$ construct a path from $x+yi$ by iteratively moving one step up, left, down, and left until reaching the line $x = -b/2$ ($x = -b/2 + 1$) if $x + yi$ is even (odd).

If b is odd, given our new definition of odd and even nodes the procedure for constructing the spanning tree is the same as above with some modifications. In the right region the construction of the paths continues until reaching the line $x = (b-1)/2$. In the left region the construction of the paths starts when $x < -1$ (for a given $x + yi$ node in the region) and continues until reaching the line $x = -(b-1)/2$.

Figure 4.16 shows an example of a spanning tree which is constructed in the square whose diagonal is 28.

Next we show that a degree-three pruned Gaussian network generated by $\alpha = a + bi$ (as well as a Gaussian network) can be placed within the square whose diagonal is $2b$.

Consider the degree-three pruned Gaussian network G_α where $\alpha = a + bi$ and $0 < a \leq b$. Consider the point $P = \frac{a-b}{2} + i\frac{a+b}{2}$. Since $a+b$ is even, the coordinates of P are always integers and so P is a node. From now on in this section we consider this node to be the node 0 and the addresses of the other nodes will be adjusted

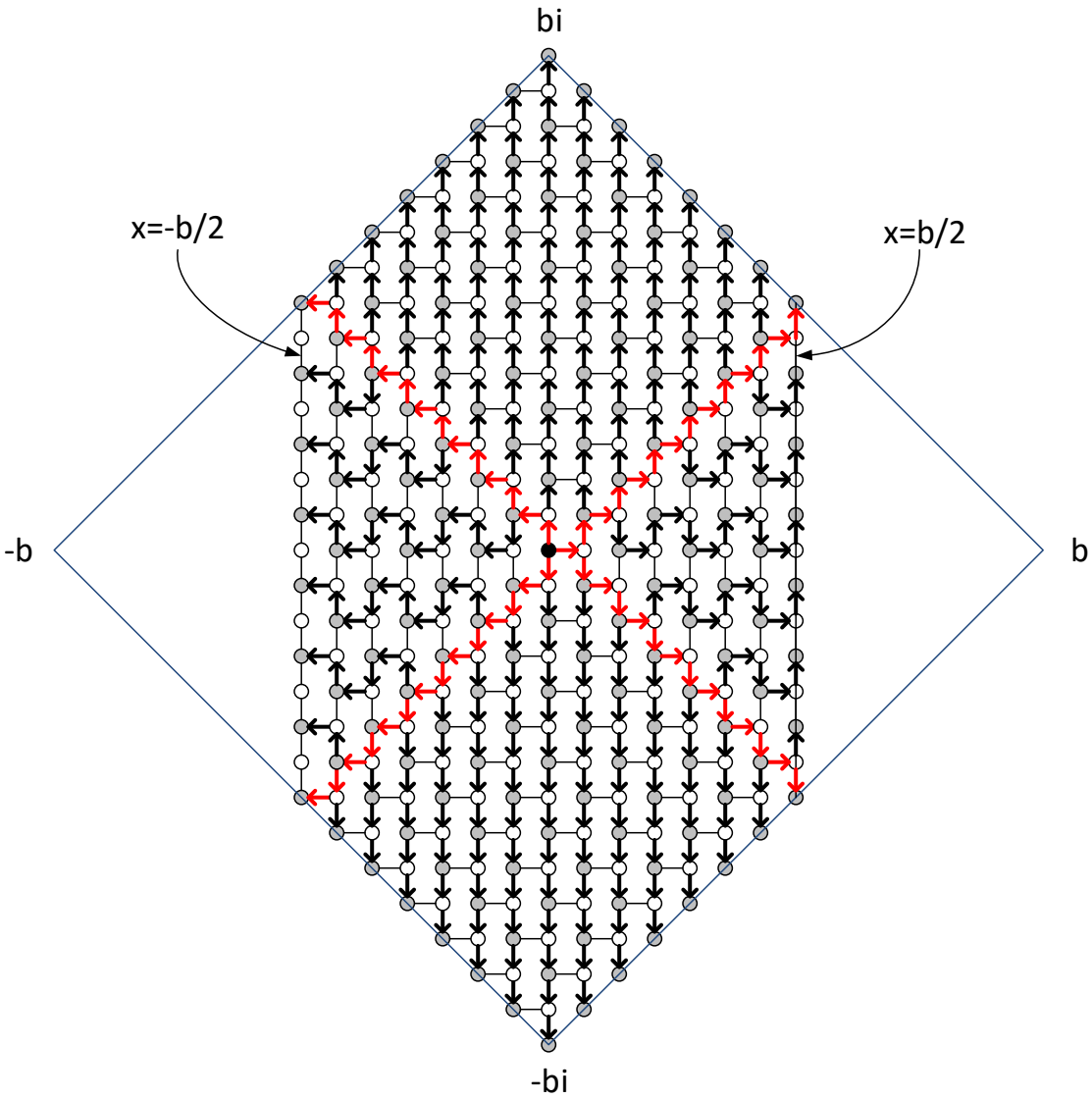


Figure 4.16: An example of a spanning tree in the square whose diagonal is 28.

accordingly. Thus, the coordinates of the corners of the basic square generated by $\alpha = a + bi$ are $\frac{a+b}{2} + \frac{b-a}{2}i$, $\frac{a-b}{2} + \frac{a+b}{2}i$, $-\frac{a+b}{2} + \frac{a-b}{2}i$, and $\frac{b-a}{2} - \frac{a+b}{2}i$. Now consider the square whose diagonal is $2b$ and its center is the node 0. This square and the basic square for G_{a+bi} are shown in Figure 4.17. It can be verified that each corner of the basic square lies on a side of the square. For example consider the side $x + y = b$ (this is the side between the nodes bi and b.) The corner $\frac{a+b}{2} + \frac{b-a}{2}i$ lies on this side because

$$\frac{a+b}{2} + \frac{b-a}{2} = b.$$

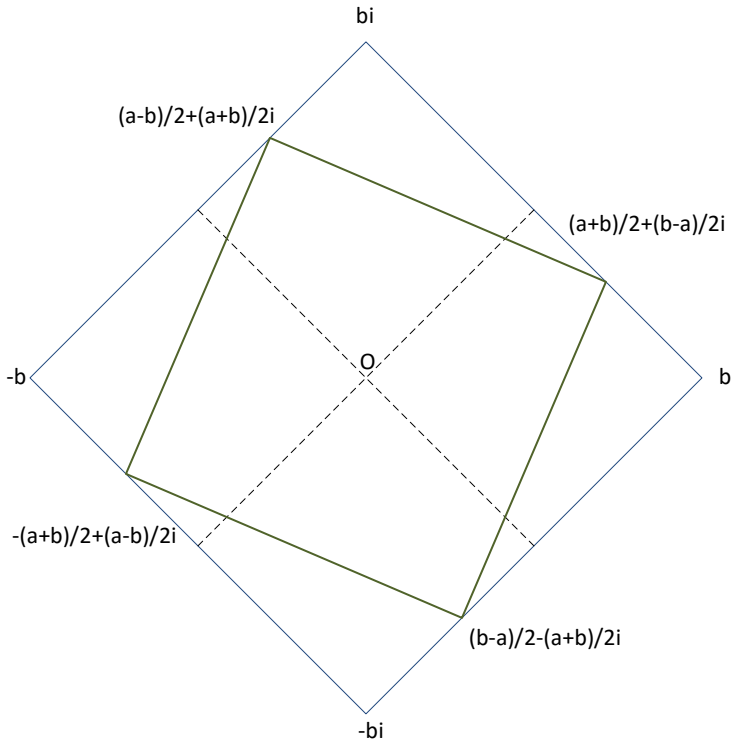


Figure 4.17: The square with diagonal $2b$ and a basic square for G_{a+bi} centered at O.

4.5.4.2 Decomposed Square Representation when $a \leq \lfloor b/2 \rfloor$

A *decomposed square*, denoted by $\mathcal{P}_d(\alpha)$, is obtained by decomposing a basic square of the degree-three pruned Gaussian network generated by $\alpha = a+bi$. That is, the basic square is divided into some nonoverlapping regions and then these regions are reassembled to form $\mathcal{P}_d(\alpha)$.

Figure 4.18 shows a basic square centered at 0 for the degree-three pruned Gaussian network G_α generated by $\alpha = a + bi$ where $0 < a < b/2$ and the square whose diagonal is $2b$ (as discussed earlier). The node O is both the center of G_α and the center of the square and it is considered to be the origin. The dashed lines in the figure represent the two lines $y = x$ and $y = -x$. The distance of the nodes on the line $x - y = -a$ from $i\alpha$ is b by (4.1) if b is even or by (4.2) if b is odd. Similarly, the distance of the nodes on the line $x - y = a$ from $i^3\alpha$ is b.

Table 4.1 shows the coordinates of the nodes and the corresponding nodes in the decomposed square $\mathcal{P}_d(\alpha)$. Note that if b is odd, then the coordinates of some of the points in Table 4.1 are not integers. But this doesn't affect our discussion because we are considering the Gaussian integer within the regions. If b is odd then there is no Gaussian integer on the lines in some of the regions.

Table 4.1: The coordinates of the nodes in Figure 4.18.

A_1	$-\frac{b}{2} + \frac{b}{2}i$	A_2	$\frac{b}{2} + \frac{b-2a}{2}i$
B_1	$-\frac{b}{2} + \frac{a}{2}i$	B_2	$\frac{b}{2} - \frac{a}{2}i$
C_1	$-\frac{b}{2} + \frac{2a-b}{2}i$	C_2	$\frac{b}{2} - \frac{b}{2}i$
E_1	$-\frac{a+b}{2} + \frac{a-b}{2}i$	E_2	$\frac{b-a}{2} - \frac{a+b}{2}i$
F_1	$\frac{a-b}{2} + \frac{a+b}{2}i$	F_2	$\frac{a+b}{2} + \frac{b-a}{2}i$
C_3	$\frac{2a-b}{2} + \frac{2a+b}{2}i$	A_5	$\frac{b-2a}{2} - \frac{b+2a}{2}i$

Consider the two triangles $A_1B_1F_1$ and $A_2B_2F_2$. Since $A_1 - A_2 = B_1 - B_2 = F_1 - F_2 = i\alpha$ all the Gaussian integers in $A_1B_1F_1$ and $A_2B_2F_2$ are congruent modulo α. Similarly, all the Gaussian integers in $B_1C_1E_1$ and $B_2C_2E_2$ are congruent modulo α. Therefore, $A_2B_2F_2$ and $B_1C_1E_1$ in $\mathcal{P}(\alpha)$ are replaced with $A_1B_1F_1$ and

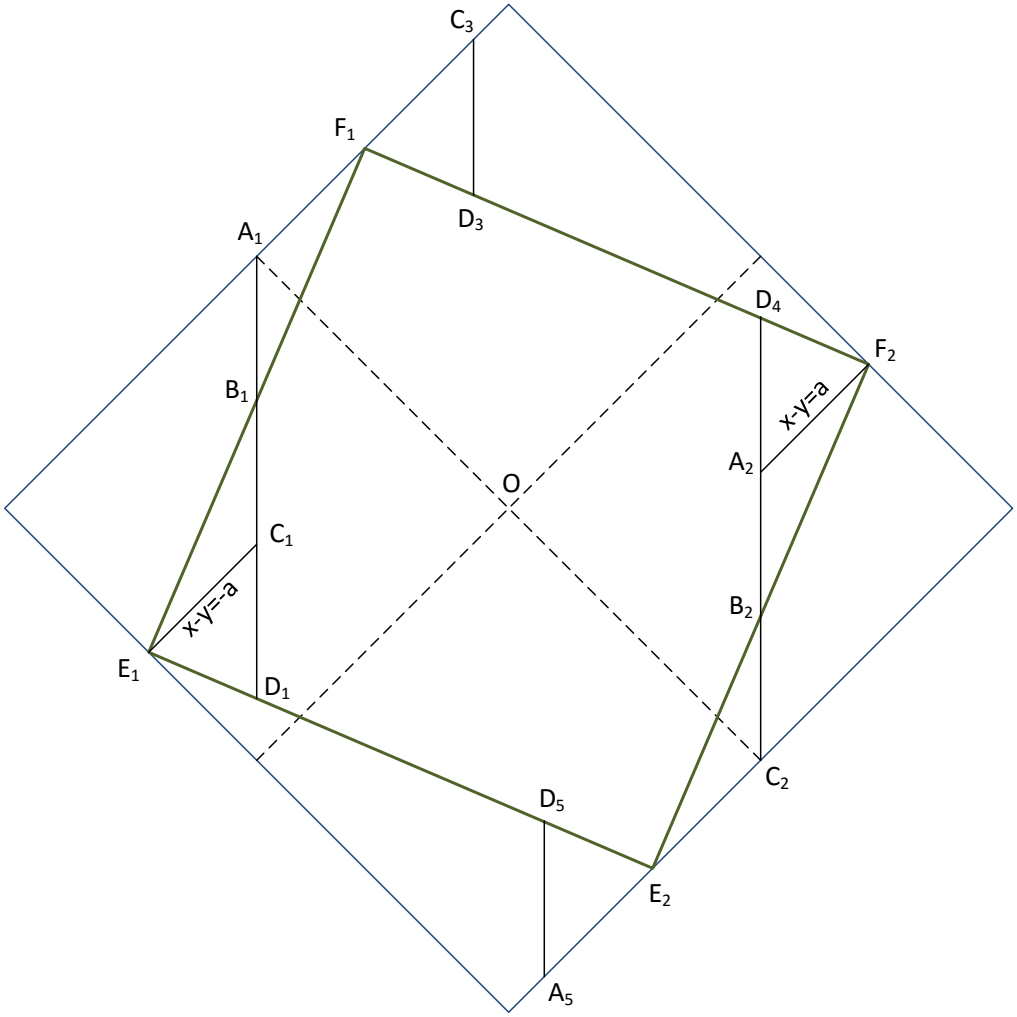


Figure 4.18: Dividing the basic square of G_{a+bi} into regions when $a \leq \lfloor b/2 \rfloor$.

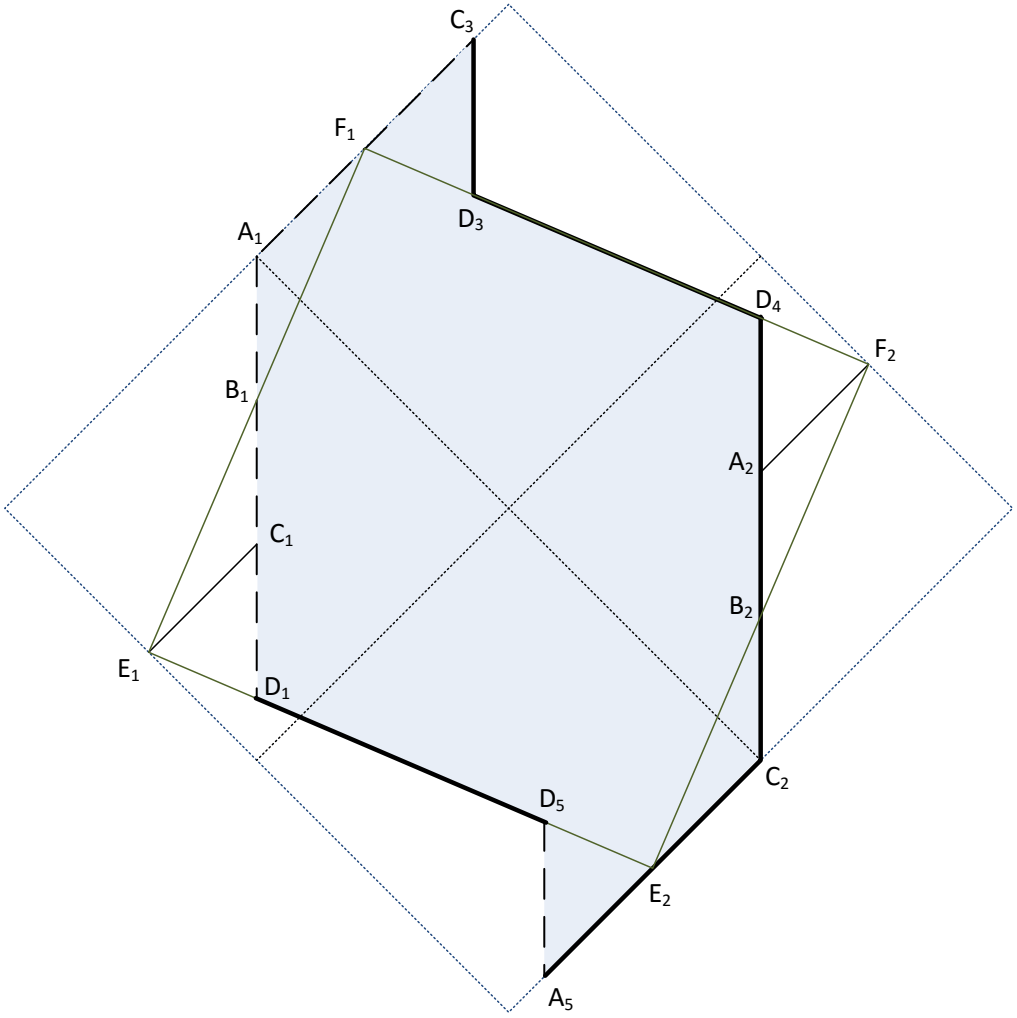


Figure 4.19: The decomposed square representation when $a \leq \lfloor b/2 \rfloor$.

$B_2C_2E_2$ in $\mathcal{P}_d(\alpha)$, respectively.

Now, consider the two triangles $C_1E_1D_1$ and $C_3F_1D_3$. We have $C_3 - C_1 = F_1 - E_1 = \alpha$. Note that $D_1 = -b/2 + y_1 i$ and $D_3 = a - b/2 + y_3 i$. In general y_1 and y_3 are not integers. However, $D_3 - D_1 = a + (y_3 - y_1)i$. y_1 and y_3 can be found by writing the equations of the lines E_1E_2 and F_1F_2, respectively. It can be easily shown that $y_3 - y_1 = b$. That means $D_3 - D_1 = a + bi = \alpha$. Thus, all the Gaussian integers in $C_1E_1D_1$ and $C_3F_1D_3$ are congruent modulo α.

Similarly, we can show that all the Gaussian integers in $A_2F_2D_4$ and $A_5E_2D_5$ are congruent modulo α. Therefore, $A_2F_2D_4$ and $C_1E_1D_1$ in $\mathcal{P}(\alpha)$ are replaced with $A_5E_2D_5$ and $C_3F_1D_3$ in $\mathcal{P}_d(\alpha)$, respectively. The decomposed square $\mathcal{P}_d(\alpha)$ is shown in Figure 4.19.

In this figure the thick solid lines represent the borders that are included in the shaded region whereas the dashed lines show the borders that are not included in the shaded region. The reason why some of the lines are not included in the shaded region is as follows. First, remember that only the nodes on the two sides (side E_1E_2 and E_2F_2) of the basic square are included in the node set of the network. The line A_1D_1 is not included because the distance of some of the nodes on this line from 0 is greater than b. Instead of the nodes on this line, the nodes on the lines A_2C_2 and C_3D_3 are considered. The line A_5D_5 is not included because the line A_2D_4 is included. The line A_1C_3 is excluded since its congruent modulo α (line A_5C_2) is included.

Note that in this section $0 < a < b/2$. In this case all the points shown in Figure 4.18 lie as shown. Consider C_3 as an example. The Y coordinate of C_3 is $a + b/2$ and since $a < b/2$ this point always lie on the border of the square. Same reasoning can be given for A_5.

Now, we can construct a spanning tree in the way that described earlier. Note that, the construction in each region continues until it covers the nodes in the network. The detail is as follows.

Consider the node $P = x + yi$ in the up region. If $x > a - b/2$ the construction continues until $2ax + 2by \leq a^2 + b^2$. (The equation of the line F_1F_2 is $2ax + 2by =$

a^2+b^2.) Else the construction continues until $|x|+y<b$. Similarly, if $P=x+yi$ is visited in the down region and if $x \leq b/2-a$ the construction continues until $2ax+2by \geq -(a^2+b^2)$. (The equation of the line E_1E_2 is $2ax+2by=-(a^2+b^2)$.) Else the construction continues until $|x|+|y| \leq b$. The construction in the right and left region is exactly as described earlier. Except that in the left region the nodes on the line $x=-b/2$ are not covered because the corresponding nodes are already covered.

4.5.4.3 Decomposed Square Representation when $a \geq \lceil b/2 \rceil$

Similar to the previous section a ball of radius $a \geq \lceil b/2 \rceil$ is defined as follows. First, we assume d is even. A *ball of radius* d and center at 0 is the set of nodes of distance less than or equal to d from 0 where the distance is calculated by (4.1). This ball is bounded by the square whose diagonals are $2d$. The left of right regions are bounded by the lines $x=d/2$ and $x=-d/2$, respectively. Again, the odd nodes on the line $x=-d/2$ are not included in the ball. (If d is odd the ball is defined similar to the case where b is odd.)

Figure 4.20 shows the basic square for the degree-three pruned Gaussian network G_α generated by $\alpha=a+bi$ where $a \geq \lceil b/2 \rceil$. This figure also shows the square whose diagonal is $2b$. The node O is both the center of G_α and the center of the square and it is considered to be the origin. The dashed lines in the figure represent the two lines $y=x$ and $y=-x$. The distance of the nodes on the lines $x-y=-a$ and $x+y=-a$ from $i^2\alpha$ is b (based on Equation (4.1) if b is even and based on Equation (4.2) if b is odd). Similarly, the distance of the nodes on the lines $x-y=a$ and $x+y=a$ from α is b.

Table 4.2 shows the coordinates of the nodes and the corresponding nodes in the decomposed square $\mathcal{P}_d(\alpha)$. Again if b is odd the coordinates of the points H_1, H_3, J_4, and J_5 are not integers. However, this doesn't affect our discussion.

Based on the discussion given in Section 4.5.4.2 the triangles $A_2B_2F_2$ and $B_1C_1E_1$ in $\mathcal{P}(\alpha)$ are replaced with $A_1B_1F_1$ and $B_2C_2E_2$ in $\mathcal{P}_d(\alpha)$, respectively. Consider the quadrilateral $G_1E_1D_1H_1$ in $\mathcal{P}(\alpha)$, it can be replaced with the quadri-

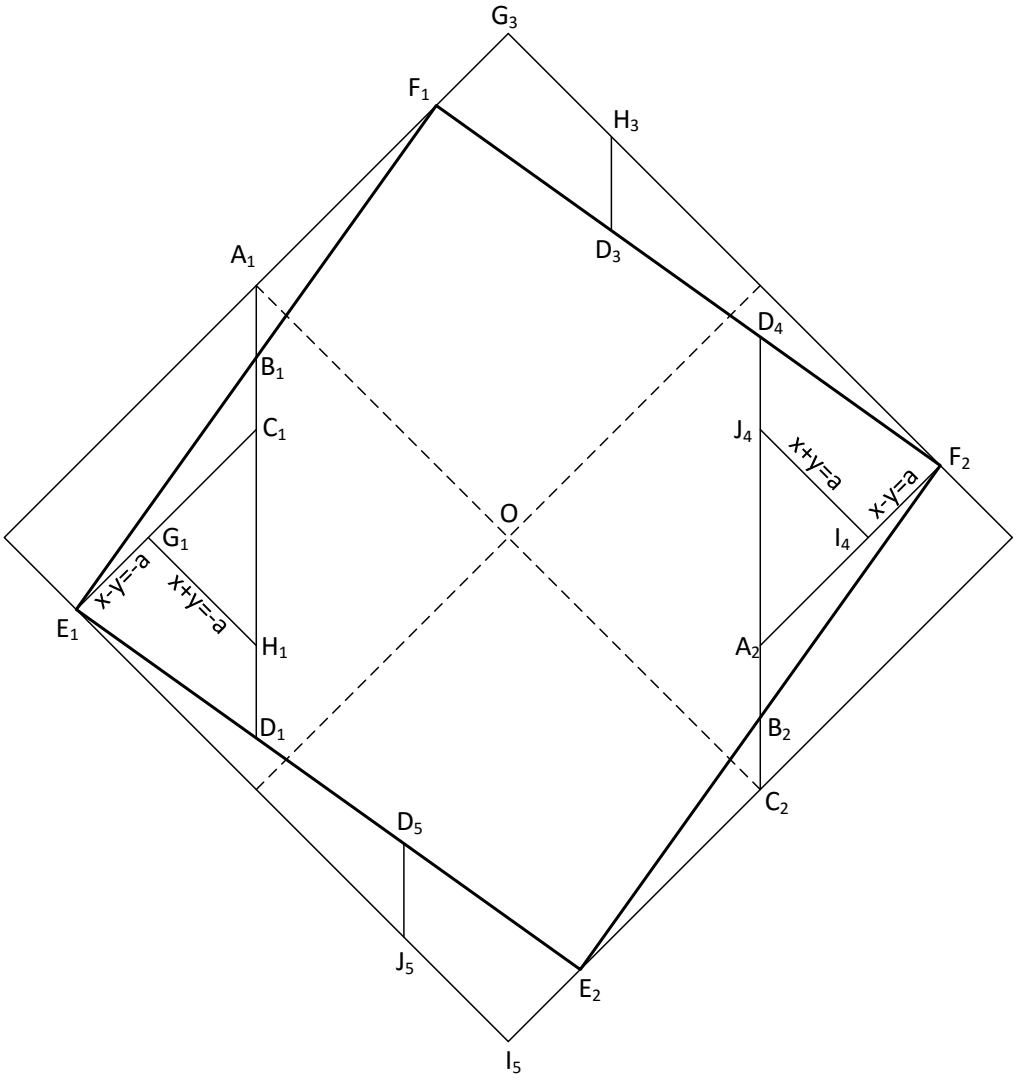


Figure 4.20: Dividing the basic square of G_{a+bi} into regions when $a > \lfloor b/2 \rfloor$.

Table 4.2: The coordinates of the nodes in Figure 4.20.

A_1	$-\frac{b}{2}+\frac{b}{2}i$	A_2	$\frac{b}{2}+\frac{b-2a}{2}i$
B_1	$-\frac{b}{2}+\frac{a}{2}i$	B_2	$\frac{b}{2}-\frac{a}{2}i$
C_1	$-\frac{b}{2}+\frac{2a-b}{2}i$	C_2	$\frac{b}{2}-\frac{b}{2}i$
E_1	$-\frac{a+b}{2}+\frac{a-b}{2}i$	E_2	$\frac{b-a}{2}-\frac{a+b}{2}i$
F_1	$\frac{a-b}{2}+\frac{a+b}{2}i$	F_2	$\frac{a+b}{2}+\frac{b-a}{2}i$
C_3	$\frac{2a-b}{2}+\frac{2a+b}{2}i$	A_5	$\frac{b-2a}{2}-\frac{b+2a}{2}i$
G_1	$-a$	G_3	bi
H_1	$-\frac{b}{2}+\frac{b-2a}{2}i$	H_3	$\frac{2a-b}{2}+\frac{3b-2a}{2}i$
I_4	a	I_5	$-bi$
J_4	$\frac{b}{2}+\frac{2a-b}{2}i$	J_5	$\frac{b-2a}{2}+\frac{2a-3b}{2}i$

lateral $G_3F_1D_3H_3$ in $\mathcal{P}_d(\alpha)$. This is because $G_3 - G_1 = F_1 - E_1 = D_3 - D_1 = H_3 - H_1 = \alpha$ and the corresponding Gaussian integers in $G_3F_1D_3H_3$ and $G_1E_1D_1H_1$ are congruent modulo α.

Similarly, consider the two quadrilaterals $I_4J_4D_4F_2$ and $I_5J_5D_5E_2$. Again, since $I_4 - I_5 = J_4 - J_5 = D_4 - D_5 = F_2 - E_2 = \alpha$ the corresponding Gaussian integers in the two quadrilateral are congruent modulo α. Thus, $I_4J_4D_4F_2$ in $\mathcal{P}(\alpha)$ is replaced with $I_5J_5D_5E_2$ in $\mathcal{P}_d(\alpha)$.

Then, $\mathcal{P}_d(\alpha)$ is given in Figure 4.21. Now, consider the two triangles $C_1G_1H_1$ and $I_4A_2J_4$. By (4.1) or (4.2) these triangles are in fact the regions where the diameter node lies, because they are the only regions where the distance from the origin (and also from α and $i\alpha$) is greater than b. We further decompose these triangles.

Figure 4.22(a) shows the triangle $C_1G_1H_1$ divided into different regions. Assume the diameter d of the network is $b+c$ for some integer $c > 0$. Let $K_1 = G_1 + c$, $L_1 = C_1 - ci$ and $M_1 = H_1 + ci$. We assign the common edges of the three regions shown in Figure 4.22(a) to each one of the regions as follows. The lines L_1K_1 and K_1M_1 except for the point K_1 belongs to the triangle $L_1K_1M_1$. The line G_1K_1

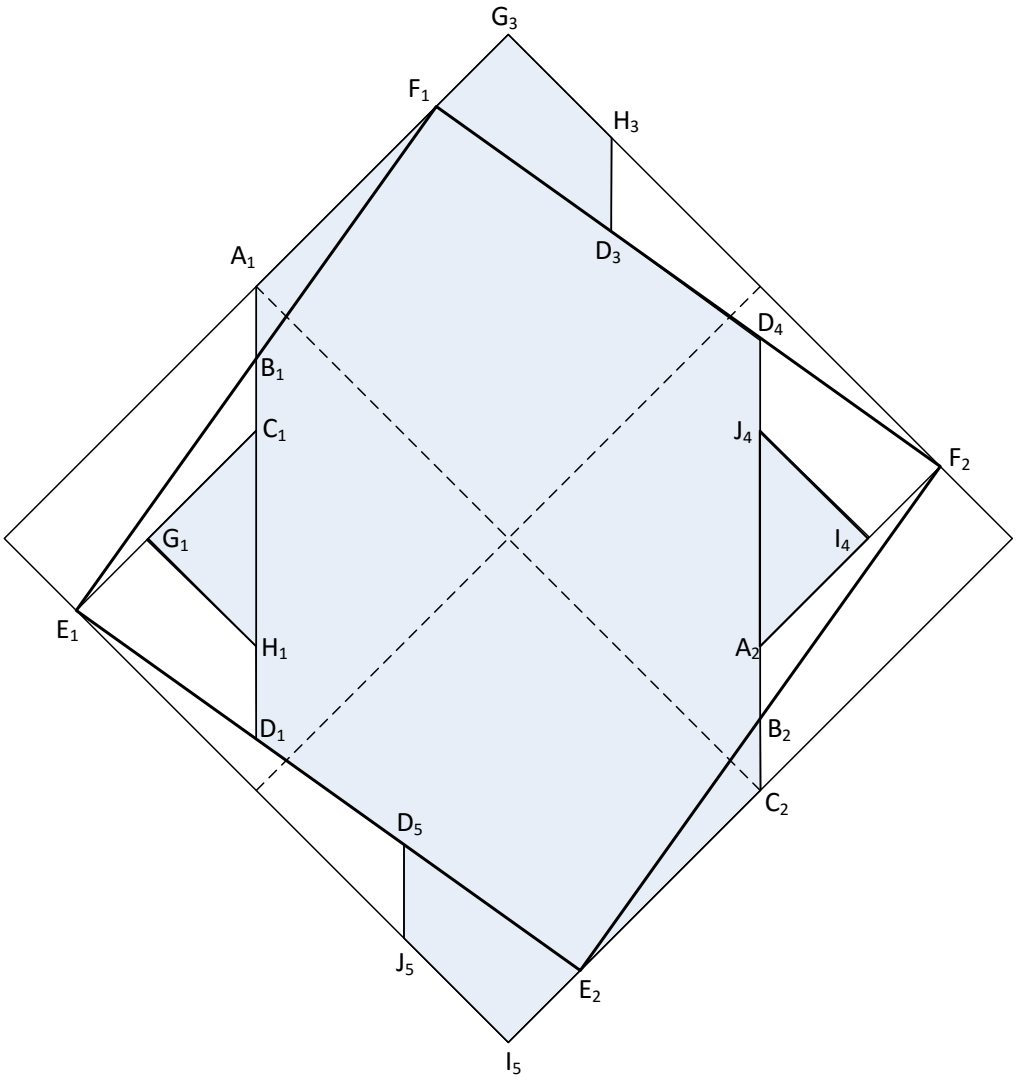


Figure 4.21: The first stage at decomposition.

belongs to region 1.

The triangle $I_4J_4A_2$ is also divided into three different regions similar to the triangle $C_1G_1H_1$ as shown in Figure 4.22(b). Here, let $K_2 = I_4 - c$, $M_2 = J_4 - ci$, and $L_2 = A_2 + ci$. The lines L_2K_2 and K_2M_2 except for the point K_2 belongs to the triangle $L_2K_2M_2$. The line I_4K_2 belongs to region 3.

Now we replace the regions 1, 2, 3, and 4 with the corresponding quadrilaterals shown in Figure 4.23. Again, reasons for the inclusion or exclusion of each line can be given.

It can be verified that the Gaussian integers within regions 1, 2, 3, and 4 in Figure 4.23 and the corresponding regions in Figure 4.22(a) and Figure 4.22(b) are congruent modulo α. For example, the points in region 1 in Figure 4.22(a) and the points in region 1 in Figure 4.23 are congruent modulo α because $C_1 - C_2 = i\alpha$ (as well as the other three corners of the two quadrilaterals).

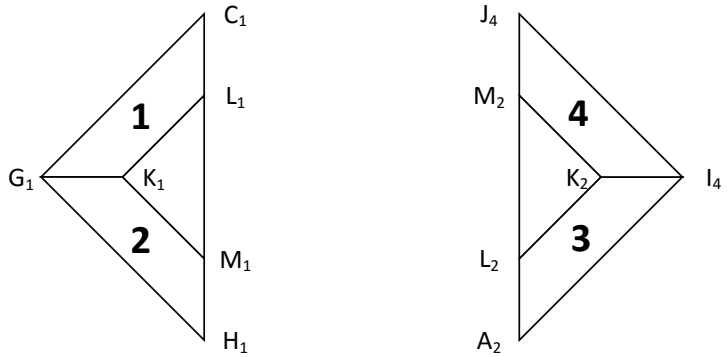


Figure 4.22: The two regions where diameter nodes may lie.

The spanning tree for the $\mathcal{P}_d(\alpha)$ is constructed as described earlier. Note that, the construction in each region continues until it covers the nodes in the network. The detail is as follows.

Consider the node $P = x + yi$ in the up region. If $x > a - b/2$ the construction continues until $2ax + 2by \leq a^2 + b^2$. (The equation of the line F_1F_2 is $2ax + 2by = a^2 + b^2$.) Else if $a - b < x < 0$ the construction continues until $|x| + y < b$. For the

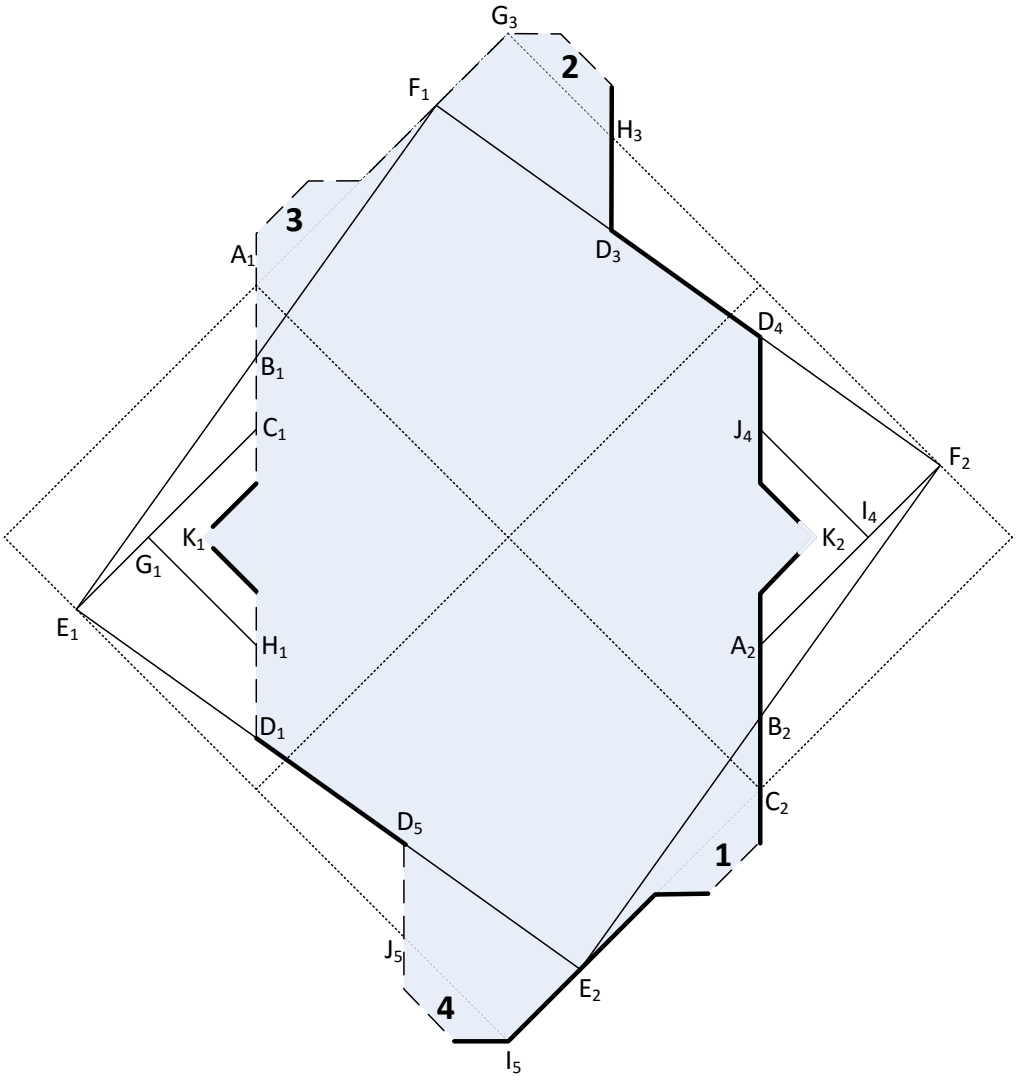


Figure 4.23: The final layout of the decomposed square representation when $a > \lfloor b/2 \rfloor$.

remaining part (that is $0 < x < a - b/2$ and $x < a - b$) the construction continues until $|x| + y < d$.

Similarly, if $P = x + yi$ is visited in the down region and if $x \leq b/2 - a$ the construction continues until $2ax + 2by \geq -(a^2 + b^2)$. (The equation of the line E_1E_2 is $2ax + 2by = -(a^2 + b^2)$.) Else if $0 < x < a - b$ the construction continues until $|x| + |y| \leq b$. For the remaining part (that is $b/2 - a < x < 0$ and $x > a - b$) the construction continues until $|x| + |y| < d$.

The construction in the right and left region is the same as described earlier and continues in the right region until it covers the nodes on the line $x = b/2$. The construction continues in the left region up to but not including the nodes on the line $x = -b/2$. In addition in both regions if $b/2 - a + c \leq y \leq a - b/2 + c$ (recall that we assume d=b+c) then the construction continues unit it covers the nodes within the two triangles $K_1L_1M_1$ and $K_2L_2M_2$.

We next prove that the length of any broadcast from 0 is at most d (diameter of the network). This can be seen for the up and down regions because these regions are bounded by the lines $|x| + |y| = d$. Consider the left region. The broadcasts in this region stop at the node $K_1 + 1$ because node K_1 is covered by region 1. Let l be X offset of the node $K_1 + 1$ from 0. The value for l is found by subtracting the length of the line G_1O from the length of the line G_1K_1 minus one; that is, $l = a - c - 1$. By (4.1) the length of any path from 0 to $x = K_1 + 1$ in the left region is at most $2l + 1$. Thus, we need to show that $2l + 1 \leq 2k + r$ where $a + b = 3k + r$ for $r = 0, \pm 1$. Thus, we have

$$l = a - c - 1 = a - (d - b) - 1 = a + b - d + 1 = 3k + r - (2k + r) - 1 = k - 1$$

and the distance from $K_1 + 1$ to 0 is $2(k-1) + 1 \leq 2k + r$ since $-1 \leq r$. Therefore, we have proved that any broadcast length in the left region is at most d.

Now consider the right region. The broadcasts in this region stop at the node $K_2 - 1$ because node K_2 is covered by region 3. Let l' be X offset of the node $K_2 - 1$ from 0. The value for l' is found by subtracting the length of the line I_4O from the length of the line I_4K_2 minus one; that is, $l' = a - c - 1$. This is the same

value obtained above for l, so $l = l'$. By (4.1) the length of any path from 0 to $x = K_2 - 1$ in the right region is at most $2l$. Thus, we need to show that $2l \leq 2k + r$ where $a + b = 3k + r$ for $r = 0, \pm 1$. We already showed that $2l + 1 \leq 2k + r$, so $2l \leq 2k + r$. Therefore, we have proved that any broadcast length in the right region is at most d.

4.6 Conclusion and Future Research Directions

In this chapter we introduce a class of degree-three networks obtained from pruning Gaussian networks. We prove that these networks are Cayley graphs, and thus node-symmetric. We also calculate the diameter of the network. The diameter of the degree-three pruned Gaussian network is less than the diameter of the honeycomb torus with approximately the same number of nodes. In addition, we develop some communication algorithms for the proposed network. These algorithms include a routing and a one-to-all broadcasting algorithm.

Developing a deadlock-free routing algorithm for degree-three pruned Gaussian networks is a possible future research direction.

Chapter 5: Conclusion

This thesis considers interconnection networks based on Eisenstein-Jacobi and Gaussian integers. Chapter 2 proposes two new deadlock-free and minimal routing algorithms for hexagonal meshes and tori. By prohibiting certain turns in the resource dependence graph, the proposed algorithms remain deadlock-free. Since they are minimal, no livelock can happen in the network. Additional virtual channels per physical channel are used to remove inherent cycles caused by the wraparound edges.

Chapter 3 discusses higher dimensional Gaussian networks. We show that the diameter and the average distance of a higher dimensional Gaussian network are less than those of a multidimensional torus with approximately the same number of nodes and degree. An optimal one-to-one routing algorithm has been developed. Simulation results show that our routing algorithm does outperform the usual routing algorithm for tori in terms of average message latency. We extend many properties of one-dimensional Gaussian networks to the higher dimensional setting, including the distance distribution and an optimal one-to-all broadcasting algorithm. In particular, when $n = 2^r$ we generate $2n$ edge-disjoint Hamiltonian cycles in n-dimensional Gaussian networks generated by $\alpha = a + bi$ for nonnegative integers a, b where $\gcd(a, b) = 1$. We have also shown how to embed multi-dimensional edge-disjoint torus networks on this network.

Chapter 4 proposes a degree-three interconnection network that is obtained from pruning a Gaussian network. This network shows possible performance improvement over the other degree-three networks such as honeycomb rectangular tori in that it has smaller diameter. A routing algorithm is given in this chapter.

A possible future research direction is to obtain fully adaptive and deadlock free routing algorithms for the three main families of interconnection networks discussed in this thesis. In addition, developing fault-tolerant routing algorithms

(that is, a routing algorithm that can route a packet in the existence of faulty nodes or edges) would be another direction for research.

Bibliography

[1] Yuuichirou Ajima, Tomohiro Inoue, Shinya Hiramoto, and Toshiyuki Shimizu. Tofu: Interconnect for the K computer. *FUJITSU Science Technology*, 48(3):280–285, July 2012.

[2] B. Albader, B. Bose, and M. Flahive. Efficient communication algorithms in hexagonal mesh interconnection networks. *IEEE Transactions on Parallel and Distributed Systems*, 23(1):69–77, January 2012.

[3] R. Alverson, D. Roweth, and L. Kaplan. The Gemini system interconnect. In *2010 IEEE 18th Annual Symposium on High Performance Interconnects (HOTI)*, pages 83–87, August 2010.

[4] M.M. Bae and B. Bose. Edge disjoint Hamiltonian cycles in k-ary n-cubes and hypercubes. *IEEE Transactions on Computers*, 52(10):1271–1284, 2003.

[5] Dimitri P. Bertsekas and John N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, 1997.

[6] George B.P. Bezerra, Stephanie Forrest, Melanie Forrest, Al Davis, and Payman Zarkesh-Ha. Modeling NoC traffic locality and energy consumption with Rent's communication probability distribution. In *Proceedings of the 12th ACM/IEEE International Workshop on System Level Interconnect Prediction*, SLIP '10, pages 3–8, New York, NY, USA, 2010. ACM.

[7] B. Bose, B. Broeg, Younggeun Kwon, and Y. Ashir. Lee distance and topological properties of k-ary n-cubes. *IEEE Transactions on Computers*, 44(8):1021–1030, 1995.

[8] B. Bose, A. Shamaei, and M. Flahive. Higher dimensional Gaussian networks. *IEEE Transactions on Parallel and Distributed Systems*, PP(99):1–1, 2015.

[9] J. Carle and J.F. Myoupo. Topological properties and optimal routing algorithms for three dimensional hexagonal networks. In *The Fourth International Conference/Exhibition on High Performance Computing in the Asia-Pacific Region, 2000. Proceedings*, volume 1, pages 116–121 vol.1, May 2000.

[10] 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 26:1–26:10, New York, NY, USA, 2011. ACM.

[11] M.-S. Chen, K.G. Shin, and D.D. Kandlur. Addressing, routing, and broadcasting in hexagonal mesh multiprocessors. *IEEE Transactions on Computers*, 39(1):10 –18, January 1990.

[12] P. Christie and D. Stroobandt. The interpretation and application of Rent's rule. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 8(6):639–648, December 2000.

[13] Martin Cohn. Affine m-ary Gray codes. *Information and Control*, 6(1):70–78, March 1963.

[14] Zarka Cvetanovic. Performance analysis of the Alpha 21364-based HP GS1280 multiprocessor. *SIGARCH Comput. Archit. News*, 31(2):218–229, May 2003.

[15] W. J Dally and C. L Seitz. Deadlock-free message routing in multiprocessor interconnection networks. *IEEE Transactions on Computers*, C-36(5):547–553, May 1987.

[16] William J. Dally and Charles L. Seitz. The torus routing chip. *Distributed Computing*, 1(4):187–196, December 1986.

[17] William J. Dally and Brian Towles. *Principles and practices of interconnection networks*. Morgan Kaufmann, 2004.

[18] Satish Damaraju, Varghese George, Sanjeev Jahagirdar, Tanveer Khondker, R. Milstrey, Sanjib Sarkar, Scott Siers, I. Stolero, and Arun Subbiah. A 22nm IA multi-CPU and GPU System-on-Chip. In *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2012 IEEE International*, pages 56–57, February 2012.

[19] Al Davis. Mayfly: A general-purpose, scalable, parallel processing architecture. *Lisp Symb. Comput.*, 5(1-2):7–48, May 1992.

[20] K. Day and A.E. Al-Ayyoub. The cross product of interconnection networks. *IEEE Transactions on Parallel and Distributed Systems*, 8(2):109–118, 1997.

[21] C. Decayeux and D. Seme. 3d hexagonal network: modeling, topological properties, addressing scheme, and optimal routing algorithm. *IEEE Transactions on Parallel and Distributed Systems*, 16(9):875–884, September 2005.

[22] J.W. Dolter, P. Ramanathan, and K.G. Shin. Performance analysis of virtual cut-through switching in HARTS: A hexagonal mesh multicomputer. *IEEE Transactions on Computers*, 40(6):669 –680, June 1991.

[23] J. Duato. A new theory of deadlock-free adaptive routing in wormhole networks. *IEEE Transactions on Parallel and Distributed Systems*, 4(12):1320–1331, December 1993.

[24] M. Flahive and B. Bose. The topology of Gaussian and Eisenstein-Jacobi interconnection networks. *IEEE Transactions on Parallel and Distributed Systems*, 21(8):1132–1142, August 2010.

[25] M. Flahive and B. Bose. On resource placement in Gaussian and EJ interconnection networks. *IEEE Transactions on Computers*, 62(3):623–626, March 2013.

[26] F. Garcia Nocetti, Julio Solano, Ivan Stojmenovic, and Milos Stojmenovic. Higher dimensional hexagonal networks. *Journal of Parallel and Distributed Computing*, 63(11):1164–1172, November 2003.

[27] F. Garcia Nocetti, I. Stojmenovic, and Jingyuan Zhang. Addressing and routing in hexagonal networks with applications for tracking mobile users and connection rerouting in cellular networks. *IEEE Transactions on Parallel and Distributed Systems*, 13(9):963 – 971, September 2002.

[28] C.J. Glass and L.M. Ni. The turn model for adaptive routing. In *, The 19th Annual International Symposium on Computer Architecture, 1992. Proceedings*, pages 278–287, 1992.

[29] Wim Heirman, Joni Dambre, Dirk Stroobandt, and Jan Van Campenhout. Rent's rule and parallel programs: Characterizing network traffic behavior. In *Proceedings of the 2008 International Workshop on System Level Interconnect Prediction*, SLIP '08, pages 87–94, New York, NY, USA, 2008. ACM.

[30] K Huber. Codes over Eisenstein-Jacobi integers. In *Finite Fields: Theory, Applications, and Algorithms*, volume 168, pages 165–179. American Mathematical Soc., 1994.

[31] Dilip D. Kandlur and Kang G. Shin. Reliable broadcast algorithms for HARTS. *ACM Trans. Comput. Syst.*, 9(4):374–398, November 1991.

[32] M. Karpovsky, L. Levitin, and M. Mustafa. Optimal turn prohibition for deadlock prevention in networks with regular topologies. *IEEE Transactions on Control of Network Systems*, 1(1):74–85, March 2014.

[33] Vipin Kumar, Ananth Grama, Anshul Gupta, and George Karypis. *Introduction to parallel computing: design and analysis of algorithms*. Benjamin-Cummings Publishing Co., Inc., Redwood City, CA, USA, 1994.

[34] Ding-Ming Kwai and B. Parhami. A class of fixed-degree Cayley-graph interconnection networks derived by pruning k-ary n-cubes. In *, Proceedings of the 1997 International Conference on Parallel Processing, 1997*, pages 92–95, August 1997.

[35] S Lakshmivarahan, Jung-Sing Jwo, and S. K Dhall. Symmetry in interconnection networks based on Cayley graphs of permutation groups: A survey. *Parallel Computing*, 19(4):361–407, April 1993.

[36] C. Martinez, R. Beivide, E. Stafford, M. Moreto, and E.M. Gabidulin. Modeling toroidal networks with the Gaussian integers. *IEEE Transactions on Computers*, 57(8):1046 –1056, August 2008.

[37] C. Martinez, E. Stafford, R. Beivide, and E. Gabidulin. Modeling hexagonal constellations with Eisenstein-Jacobi graphs. *Problems of Information Transmission*, 44(1):1–11, 2008.

[38] Carmen Martinez, Enrique Vallejo, Ramn Beivide, Cruz Izu, and Miquel Moret. Dense Gaussian networks: Suitable topologies for on-chip multiprocessors. *International Journal of Parallel Programming*, 34:193–211, 2006.

[39] Carmen Martinez, Enrique Vallejo, Miquel Moreto, Ramon Beivide, and Mateo Valero. Hierarchical topologies for large-scale two-level networks. XVI Jornadas de Paralelismo, September 2005.

[40] A. Nayebi, S. Meraji, A. Shamaei, and H. Sarbazi-Azad. XMulator: A listener-based integrated simulation platform for interconnection networks. In *First Asia International Conference on Modeling Simulation, 2007. AMS '07*, pages 128 –132, March 2007.

[41] B. Parhami and Ding-Ming Kwai. A unified formulation of honeycomb and diamond networks. *IEEE Transactions on Parallel and Distributed Systems*, 12(1):74–80, 2001.

[42] Arash Shamaei, Bella Bose, and Mary Flahive. Higher dimensional Gaussian networks. In *Parallel Distributed Processing Symposium Workshops (IPDPSW), 2014 IEEE International*, pages 1438–1447, May 2014.

[43] I.A. Stewart. Interconnection networks of degree three obtained by pruning two-dimensional tori. *IEEE Transactions on Computers*, 63(10):2473–2486, October 2014.

[44] I. Stojmenovic. Honeycomb networks: Topological properties and communication algorithms. *IEEE Transactions on Parallel and Distributed Systems*, 8(10):1036–1042, 1997.

[45] E. Vallejo, R. Beivide, and C. Martinez. Practicable layouts for optimal circulant graphs. In *13th Euromicro Conference on Parallel, Distributed and Network-Based Processing, 2005. PDP 2005*, pages 118–125, February 2005.

[46] Dong Xiang and Wei Luo. An efficient adaptive deadlock-free routing algorithm for torus networks. *IEEE Transactions on Parallel and Distributed Systems*, 23(5):800 –808, May 2012.

[47] Wenjun Xiao and Behrooz Parhami. Further mathematical properties of Cayley digraphs applied to hexagonal and honeycomb meshes. *Discrete Applied Mathematics*, 155(13):1752–1760, August 2007.

[48] A. Youssef. Design and analysis of product networks. In *Frontiers of Massively Parallel Computation, 1995. Proceedings. Frontiers '95., Fifth Symposium on the*, pages 521–528, February 1995.

[49] Zhemin Zhang, Zhiyang Guo, and Yuanyuan Yang. Efficient all-to-all broadcast in Gaussian on-chip networks. *IEEE Transactions on Computers*, 62(10):1959–1971, October 2013.