One-to-Many Node Disjoint Paths Routing in Gaussian Interconnection Networks - Ver.9 5/15/13

Omar Alsaleh, Bella Bose and Bechir Hamdaoui

School of Electrical and Computer Engineering
Oregon State University
1148 Kelley Engineering Center
Corvallis, OR 97331, USA
Email: {alsaleh,bose,hamdaoui}@eecs.orst.edu

The cores in parallel systems are connected using interconnection networks. In this work, we consider the Gaussian interconnection networks. Unlike meshes and tori, Gaussian interconnection networks are regular grid-like networks with low degree that can accommodate more nodes with less communication latency. We first design an efficient algorithm with a constant time complexity that constructs node disjoint paths (NDP) from a single source node to the maximum number of destination nodes in the Gaussian interconnection networks without depending on the network size. Then, we prove that this algorithm always returns a solution. Also, we derive the lower and upper bounds of the sum of the NDP lengths and the time complexity of the algorithm. Finally, we show via simulation that on the average 10% more hops than the sum of shortest paths are required to construct the NDP.

Keywords: parallel processing, Gaussian interconnection network, routing, node disjoint paths.

Received May 20, 2013; revised June 20, 2013

1. INTRODUCTION

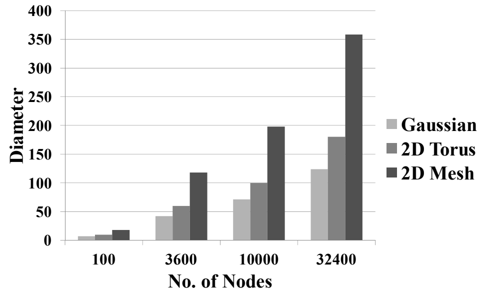
Since the switching speed of the VLSI systems is approaching the maximum limit, parallel systems play important role in improving the system performance by exploiting the inherent parallelism in problems. In the last decade, supercomputers with thousands of processors have been built - for example Cray Jaguar [1], IBM BlueGene [2], etc. These processors are linked to each other to form an interconnection network where each node represents a processor.

Achieving high computing performance critically depends on the interconnection networks. Designers of the interconnection networks seek desirable attributes such as low node degree, small diameter, and strong fault tolerance to maximize the computing performance [3–5]. Parallel computing performance depends primarily on the network topology which describes how the nodes are interconnected. As a result, many different topologies have been investigated extensively in the literature with the main objective of studying and assessing which ones yield the best computing performance [6–11]. The probability of failure in delivering the massages between the processors directly affects the computing performance. This probability can become higher because of the continuous increase in the number of processors. Therefore, it is critical to construct mutually node disjoint paths (NDP) in order to establish communication routes under such a faulty environment. Solving NDP problems for one-to-one, one-to-many, and many-to-many is fundamental and essential for ensuring fault tolerance in parallel systems.

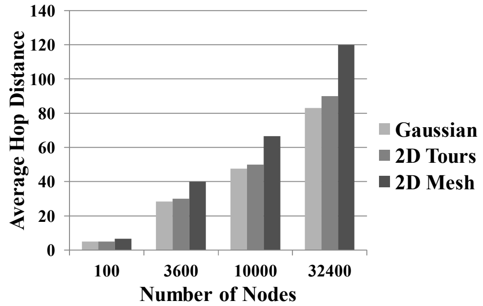
The one-to-many NDP routing problem is described as follows: given a source node s, a set of distinct destination nodes $T = \{t_1, t_2, \ldots, t_\ell\}$, where $s \notin T$ and ℓ is the node degree, construct ℓ NDP such that: 1) each path connects the source node s with one of the destination nodes $t_j \in T$, $j \in \{1, 2, \ldots, \ell\}$, and 2) the only common node along all paths is the source node s.

The NDP problems have been studied for different interconnection networks. The following related works are some examples [12–29]:

- **One-to-One NDP:** This problem has been solved for the following interconnection networks: Hierarchical Hypercube [12], k-ary n-cube [19], Hypercube [20], and (n, k)-Star [21].
- **Many-to-Many NDP:** This problem has been solved for the following interconnection networks:



(a) Diameter



(b) Average hop distance

FIGURE 1: Comparing Gaussian networks with Mesh and Tours networks

Hierarchical Hypercube [14], Metacube [28], Dual-Cubes [29], and Hypercube [18].

- **One-to-Many NDP:** This problem has been solved for the following interconnection networks: Hierarchical Hypercube [13], Dual-Cubes [15], Metacube [16], Folded Hypercube [17], Biswapped [23], Hypercube in optimal time [24], Hyper-Star [25], k-ary n-cube [30], Rotator graphs [26], and pancake graphs [27].

Unlike those previous works, we solve the problem of routing from a single source node to the maximum number of destination nodes (one-to-many) in Gaussian networks using node disjoint paths (NDP).

Gaussian networks have significant topological advantages over traditional mesh and tours networks in terms of diameter and average hop distance [31, 32]. Figure1 compares Gaussian networks with the 2-dimensional mesh and torus networks in terms of the diameter (Figure1a) and the average hop distance (Figure1b). Clearly, the diameter and average hop distance of Gaussian networks are less than the diameter and average hop distance of meshes and tori with the same number of nodes. This means Gaussian networks can accommodate more nodes with less communication latency while maintaining regular grid-like graphs.

The contributions of this work are:

1. proposing an efficient algorithm to solve the one-to-many NDP routing problem in Gaussian networks without depending on the network size,
2. theoretically proving that the proposed algorithm always returns a solution,
3. theoretically proving that the sum of NDP lengths from the source node to the destination nodes constructed by the proposed algorithm is bounded between the sum of the shortest paths and the this sum plus $(6k - 11)$ where k is the diameter,
4. analysing the time complexity to show that the time complexity of the algorithm is constant $O(1)$, and
5. simulating the algorithm to show that on the average the sum of NDP lengths is 10% more than the sum of shortest paths.

The rest of the paper is organized as follows: Section 2 recalls several preliminaries about the Gaussian networks, Section 3 describes the proposed routing algorithm, Section 4 shows the simulation results, and Section 5 concludes this paper.

2. GAUSSIAN NETWORKS PRELIMINARIES

Gaussian networks are defined in terms of Gaussian integers. The following subsections explain the Gaussian integers, describe the Gaussian networks, and formally define the one-to-many node disjoint paths (NDP) routing problem in these networks.

2.1. Gaussian Integers

A Gaussian integer is a complex number such that its real and imaginary parts are both integers. The set of all Gaussian integers $\mathbb{Z}[\mathbf{i}]$ is defined as follows:

$$\mathbb{Z}[\mathbf{i}] = \{x + y\mathbf{i} | x, y \in \mathbb{Z}\}.$$

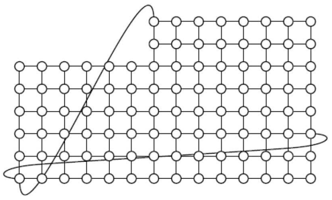
The set $\mathbb{Z}[\mathbf{i}]$ is an Euclidean domain and the norm of a Gaussian integer $\omega = \omega_x + \omega_y\mathbf{i}$ is defined as follows [31]:

$$\mathcal{N}(\omega) = {\omega_x}^2 + {\omega_y}^2.$$

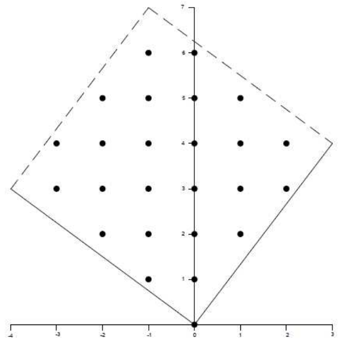
So, an Euclidean division algorithm for Gaussian integers exists. Let $\omega_1, \omega_2 \in \mathbb{Z}[\mathbf{i}]$ and $\omega_2 \neq 0$. Then, there exist $q, r \in \mathbb{Z}[\mathbf{i}]$ such that $\omega_1 = q\omega_2 + r$ and $\mathcal{N}(r) < \mathcal{N}(\omega_2)$ [31]. Let $\alpha = a + b\mathbf{i} \in \mathbb{Z}[\mathbf{i}]$ be nonzero where a and b are integers. Then, $\omega_1, \omega_2 \in \mathbb{Z}[\mathbf{i}]$ are congruent modulo α if there exists $\gamma \in \mathbb{Z}[\mathbf{i}]$ such that $\omega_2 - \omega_1 = \gamma\alpha$. Congruence and the Gaussian integers modulo α are denoted by $\omega_2 \equiv \omega_1(\text{mod } \alpha)$ and $\mathbb{Z}[\mathbf{i}]_\alpha$, respectively. The number of elements in $\mathbb{Z}[\mathbf{i}]_\alpha$ is equal to $\mathcal{N}(\alpha) = a^2 + b^2$ [31]. For example, if $a = 1$ and $b = 2$, then $\alpha = 1 + 2\mathbf{i}$ and $\mathbb{Z}[\mathbf{i}]_\alpha$ has $\mathcal{N}(1 + 2\mathbf{i}) = 1^2 + 2^2 = 5$ elements.

2.2. Gaussian Networks

Gaussian networks are two-dimensional networks generated by Gaussian integers [31, 32]. Let $\alpha \in \mathbb{Z}[\mathbf{i}]$



(a) Two adjacent meshes ($\alpha = 6 + 8\mathbf{i}$)



(b) Leaned square ($\alpha = 3 + 4\mathbf{i}$)

FIGURE 2: Different representations of Gaussian networks

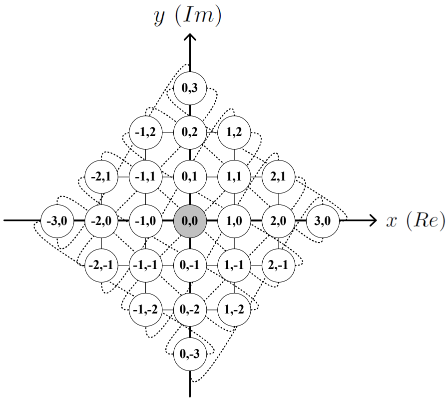


FIGURE 3: G_3 ($\alpha = 3 + 4\mathbf{i}$)

be nonzero. Each node address in a Gaussian network generated by α is a Gaussian integer that belongs to the Gaussian integers modulo α denoted by $\mathbb{Z}[\mathbf{i}]_\alpha$. So, the number of nodes in this Gaussian network is equal to $\mathcal{N}(\alpha)$. These nodes can be represented in several ways. One representation is by placing the nodes on two adjacent square meshes (see Figure2a) [32, 33]. Another representation is by placing the nodes on a leaned square (see Figure2b) [31, 33]. In this work, we use different representation which is explained in [34] (see Figure3). In this representation, the nodes are placed on a two-dimensional Cartesian plan where the x-axis and y-axis represent the real and imaginary parts of each node respectively.

For a specific integer $k \in \mathbb{Z}^+$, the following theorem shows how to get the largest Gaussian network in terms of the number of nodes. Its proof is given in [33].

Theorem 2.1 ([33]). *For a given integer $k \in \mathbb{Z}^+$, Gaussian network achieves the largest network size with $k^2 + (k+1)^2$ nodes when it is generated by $\alpha = k + (k+1)\mathbf{i}$.*

The Gaussian network generated by $\alpha = k + (k+1)\mathbf{i}$ has the shortest network diameter k among all Gaussian networks with the same number of nodes. In this work, we assume the generator of the Gaussian network is $\alpha = k + (k+1)\mathbf{i}$ and denote this Gaussian network by G_k where k is the network diameter. Figure3 shows the Gaussian network G_3 generated by $\alpha = 3 + 4\mathbf{i}$. In this example, the number of nodes is equal to $\mathcal{N}(3 + 4\mathbf{i}) = 3^2 + 4^2 = 25$ and the diameter $k = 3$.

In the following, we describe some concepts and properties that are important to understand the proposed one-to-many node disjoint paths (NDP) routing in G_k.

Addressing: Each node in the Gaussian network generated by $\alpha = k + (k+1)\mathbf{i}$ is represented as $\omega = \omega_x + \omega_y\mathbf{i} \in \mathbb{Z}[\mathbf{i}]_\alpha$. For simplicity, we write $\omega = (\omega_x, \omega_y)$ to denote node ω in the network. The set of all nodes in G_k is $\{\omega = (\omega_x, \omega_y) \in \mathbb{Z} \times \mathbb{Z} \mid |\omega_x| + |\omega_y| \leq k\}$. In Figure3, the 2-tuples inside each node are the addresses.

Connectivity: Two nodes $\omega_1, \omega_2 \in \mathbb{Z}[\mathbf{i}]_\alpha$ in G_k are connected (neighbors) if and only if $(\omega_1 - \omega_2) \equiv \pm 1, \pm\mathbf{i} (\text{mod } \alpha)$ where $\alpha = k + (k+1)\mathbf{i}$ is the generator of G_k. So, each node $\omega = \omega_x + \omega_y\mathbf{i} \in \mathbb{Z}[\mathbf{i}]_\alpha$ is connected to four neighbours:

1. the north neighbor $\omega^N = \omega_x + (\omega_y + 1)\mathbf{i}$ (mod α),
2. the west neighbor $\omega^W = (\omega_x - 1) + \omega_y\mathbf{i}$ (mod α),
3. the south neighbor $\omega^S = \omega_x + (\omega_y - 1)\mathbf{i}$ (mod α), and
4. the east neighbor $\omega^E = (\omega_x + 1) + \omega_y\mathbf{i}$ (mod α)

where $\omega^N, \omega^W, \omega^S, \omega^E \in \mathbb{Z}[\mathbf{i}]_\alpha$.

The module function (mod α) is used to build the wraparound links. Let $\beta = \beta_x + \beta_y\mathbf{i} \in \{\omega_x + (\omega_y + 1)\mathbf{i}, (\omega_x - 1) + \omega_y\mathbf{i}, \omega_x + (\omega_y - 1)\mathbf{i}, (\omega_x + 1) + \omega_y\mathbf{i}\}$ be one of the neighbors before applying the module function where $\beta \notin \mathbb{Z}[\mathbf{i}]_\alpha$. Then, the module function β (mod α) is given by the following [31]:

$$\beta \ (\text{mod } \alpha) = \begin{cases} \beta - \alpha & \text{if } (\beta_x \geq 0) \wedge (\beta_y \geq 1) \\ \beta - \mathbf{i}\alpha & \text{if } (\beta_x \leq -1) \wedge (\beta_y \geq 0) \\ \beta + \alpha & \text{if } (\beta_x \leq 0) \wedge (\beta_y \leq -1) \\ \beta + \mathbf{i}\alpha & \text{if } (\beta_x \geq 1) \wedge (\beta_y \leq 0) \end{cases} \tag{1}$$

In Figure3, the dashed links are the wraparound links built using Equation 1. For example, the south neighbor of $\omega = -2 - \mathbf{i}$ is $\omega^S = -2 - 2\mathbf{i} (\text{mod } 3 + 4\mathbf{i}) = (-2 - 2\mathbf{i}) + (3 + 4\mathbf{i}) = 1 + 2\mathbf{i}$ where $\beta = -2 - 2\mathbf{i}$. Another example, the north neighbor of $\omega = -2 + \mathbf{i}$ is $\omega^N = -2 + 2\mathbf{i} (\text{mod } 3 + 4\mathbf{i}) = (-2 + 2\mathbf{i}) - \mathbf{i}(3 + 4\mathbf{i}) = (-2 + 2\mathbf{i}) + (4 - 3\mathbf{i}) = 2 - \mathbf{i}$ where $\beta = -2 + 2\mathbf{i}$.

Diameter: The diameter is the largest possible distance between any two nodes in a network. The

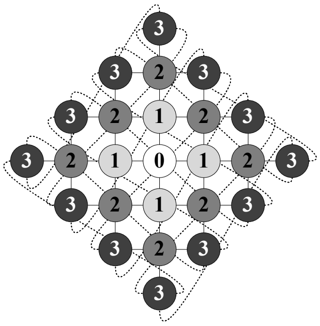


FIGURE 4: Distance distribution of G_3

diameter of G_k is equal to k [34]. For example in Figure3, the diameter of G_3 is equal to three.

Degree: The node degree is the number of its neighbors. In Gaussian networks, each node is connected to four other nodes. So, the node degree is always equal to four for all nodes [34].

Distance: The shortest distance between any two nodes $\omega_1, \omega_2 \in \mathbb{Z}[\mathbf{i}]_\alpha$ is defined as follows [32]:

$$D_\alpha(\omega_1, \omega_2) = \{|x| + |y| \ \mid (\omega_1 - \omega_2) \equiv (x + y\mathbf{i}) (\mathrm{mod}\ \alpha)\}.$$

For example in Figure3, the shortest distance between $(0, 1)$ and $(1, 2)$ is $D_\alpha((0, 1), (1, 2)) = |0 - 1| + |1 - 2| = | - 1| + | - 1| = 2$.

Since G_k is vertex symmetric, the shortest distance between node $(0, 0)$ and node $\omega = (\omega_x, \omega_y) \in \mathbb{Z}[\mathbf{i}]_\alpha$ is equal to ω's weight $W(\omega) \in \{0, 1, 2, \ldots, k\}$ which is defined as follows [32]:

$$W(\omega) = |\omega_x| + |\omega_y|. \tag{2}$$

For example in Figure3, the weight of $(1, 2)$ is $W((1, 2)) = 3$ which is the shortest distance between $(0, 0)$ and $(1, 2)$.

Based on the weight's value, Definition 2.1 defines a border node.

Definition 2.1. *Let $\omega \in \mathbb{Z}[\mathbf{i}]_\alpha$ be any node. Then, ω is a border node if and only if $W(\omega) = k$ where k is the network diameter.*

For example in Figure3, the nodes $(1, 2)$, $(3, 0)$, and $(-1, -2)$ are all border nodes.

The distance distribution of G_k gives the number of nodes $H(r)$ at distance $r \in \{0, 1, 2, \ldots, k\}$ from the $(0, 0)$ node. This distribution is defined as follows [32]:

$$H(r) = \begin{cases} 1 & \text{if} \quad r = 0 \\ 4r & \text{if} \quad 1 \leq r \leq k \end{cases}$$

For example, the distance distribution of G_3 (see Figure3), is shown in Figure4 where the number inside each node is its weight. In this example, $H(0) = 1$, $H(1) = 4$, $H(2) = 8$, and $H(3) = 12$.

Theorem 2.2 gives the shortest distance between the node $(0, 0)$ and any other node $\omega \in \mathbb{Z}[\mathbf{i}]_\alpha$ through one wraparound link as a function of its weight $W(\omega)$. We use this theorem to calculate the length of some NDP.

Theorem 2.2. *In a Gaussian network G_k, let 1) $\omega \in \mathbb{Z}[\mathbf{i}]_\alpha$ be any node such that $\omega \neq (0, 0)$ and 2) $\delta = 2(k - W(\omega)) + 1$ where k is the network degree and $W(\omega)$ the weight of ω. Then using one and only one wraparound link, the shortest distance $R(\omega)$ from node $(0, 0)$ to node ω is given as follows:*

$$R(\omega) = W(\omega) + \delta \tag{3}$$

Proof. Since any wraparound link connects two border nodes, let a and b be the border nodes that are connected using the wraparound link in the path from node $(0, 0)$ to node ω. Then using this wraparound link, the shortest distance distance from node $(0, 0)$ to the border node a and from the border node b to node $(0, 0)$ through node ω is equal to $W(a) + W(b) + 1 = 2k + 1$. Since this path goes through ω, this distance includes the distance from node $(0, 0)$ to node ω. It follows that $R(\omega) = 2k+1-W(\omega) = 2k+1-W(\omega)+W(\omega)-W(\omega) = W(\omega) + 2(k - W(\omega)) + 1 = W(\omega) + \delta$. □

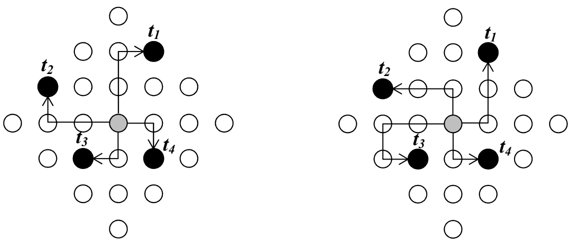
Note that δ in Equation 3 represents the extra number of hops in order to use the wraparound link. For example in Figure3, the shortest distance from node $(0, 0)$ to node $(1, 2)$ through a wraparound link is equal to $R((1, 2)) = 3 + 2(3 - 3) + 1 = 3 + 1 = 4$.

Path: A path from node ω_1 to node ω_2 is denoted by $P(\omega_1, \omega_2) = \langle \omega_1, a1, a2, \ldots, a(|P(\omega_1, \omega_2)| - 1), \omega_2 \rangle$ where $|P(\omega_1, \omega_2)|$ is the length and each two consecutive nodes (e.g. ω_1 and $a1$) along the path are neighbors. Sometimes, we write the path $P(\omega_1, \omega_2)$ as $\omega_1 \rightarrow a1 \rightarrow a2 \rightarrow \cdots \rightarrow \omega_2$.

The length of the shortest path(s) from ω_1 to ω_2 is equal to the shortest distance $D_\alpha(\omega_1, \omega_2)$ between them. For example in Figure3, one of the shortest paths between $(0, 0)$ and $(1, 1)$ is $(0, 0) \rightarrow (1, 0) \rightarrow (1, 1)$ of length two which is equal to $D_\alpha((0, 0), (1, 1)) = W((1, 1))$. Another example of a longer path $P((0, 0), (1, 1))$ is $(0, 0) \rightarrow (-1, 0) \rightarrow (-1, 1) \rightarrow (0, 1) \rightarrow (1, 1)$.

One-to-Many NDP: Given a source node s and a set of distinct destination nodes $T = \{t_j = (t_{j_x}, t_{j_y}) | 1 \leq j \leq 4\}$, where $s \notin T$, the one-to-many NDP connect s with each destination node t_j and satisfy the condition (called disjointness condition) that the only common node among all these paths is the source node s. Under this condition, the maximum number of NDP from the source node s is equal to the number of its neighbors (i.e. the node degree). Accordingly, the maximum number of destination nodes is equal to four. Since G_k is vertex symmetric, we assume the source node is $s = (0, 0)$.

For a particular set of destination nodes T, there are more than one possible NDP from s to T. One of these possibilities is denoted by $\mathbb{P}(s, T)$. For example, consider the network G_3 in Figure3, let the source node be $s = (0, 0)$ and the set of destination



(a) (b)

FIGURE 5: Different examples of NDP in G_3

nodes be $T = \{(1,2),(-2,1),(-1,-1),(1,-1)\}$. Then, one possible NDP is $\mathbb{P}(s,T) = \{\langle(0,0),(0,1),(0,2),(1,2)\rangle, \langle(0,0),(-1,0),(-2,0),(-2,1)\rangle, \langle(0,0),(0,-1),(-1,-1)\rangle, \langle(0,0),(1,0),(1,-1)\rangle\}$ (see Figure5a). Another different possibility is $\mathbb{P}(s,T) = \{\langle(0,0),(1,0),(1,1),(1,2)\rangle, \langle(0,0),(0,1),(-1,1),(-2,1)\rangle, \langle(0,0),(-1,0),(-2,0),(-2,-1),(-1,-1)\rangle, \langle(0,0),(0,-1),(1,-1)\rangle\}$ (see Figure5b).

DEFINITION 2.2. *In the Gaussian network G_k where k is the network diameter, let the source node be $s = (0,0)$ and the set of destination nodes be $T = \{t_j = (t_{j_x}, t_{j_y}) | 1 \le j \le 4\}$. Then, the sum of the lengths of the shortest distances is given by the following:*

$$L(T) = \sum_{j=1}^{4} W(t_j) \tag{4}$$

For example in Figure5a, $L(T) = 3+3+2+2 = 10$.

DEFINITION 2.3. *In the Gaussian network G_k where k is the network diameter, let the source node be $s = (0,0)$, the set of destination nodes be $T = \{t_j = (t_{j_x}, t_{j_y}) | 1 \le j \le 4\}$, and the disjoint path from the source node s to the destination node t_j be $P(s,t_j)$. Then, the sum of the lengths of the paths in $\mathbb{P}(s,T)$ is given by the following:*

$$|\mathbb{P}(s,T)| = \sum_{j=1}^{4} |P(s,t_j)| \tag{5}$$

For example in Figure5a, $|\mathbb{P}(s,T)| = 3+3+2+2 = 10$ which means the paths in $\mathbb{P}(s,T)$ are the shortest paths because $|\mathbb{P}(s,T)| = L(T)$. In Figure5b, $|\mathbb{P}(s,T)| = 3+3+4+2 = 12$.

The following section describes our routing algorithm from the source node s to each of the four destination nodes in T using NDP.

3. ONE-TO-MANY NODE DISJOINT PATHS ROUTING

The basic idea of our routing algorithm is to design a set of distinctive and comprehensive cases based on the destination nodes' locations in the network, and then construct the one-to-many node disjoint paths (NDP) $\mathbb{P}(s,T)$ for each case. The algorithm (see Alg.1) consists of two steps: case determination and NDP construction.

Alg.1 One-to-Many NDP Routing in Gaussian network G_k

Input: G_k, $T = \{t_j = (t_{j_x}, t_{j_y}) | 1 \le j \le 4\}$, $s = (s_x, s_y) \notin T$
Output: $\mathbb{P}(s,T)$

procedure *OneToMany_NDP*(G_k,T,s)
  $|Q_N| = |Q_W| = |Q_S| = |Q_E| = 0$;
  for $1 \le j \le 4$ **do** ▷ Step 1
    if $(t_{j_x} \ge s_x) \wedge (t_{j_y} \ge s_y + 1)$ **then**
      $|Q_N| = |Q_N| + 1$;
    else if $(t_{j_x} \le s_x - 1) \wedge (t_{j_y} \ge s_y)$ **then**
      $|Q_W| = |Q_W| + 1$;
    else if $(t_{j_x} \le s_x) \wedge (t_{j_y} \le s_y - 1)$ **then**
      $|Q_S| = |Q_S| + 1$;
    else
      $|Q_E| = |Q_E| + 1$;
    end if
  end for
  switch $\langle |Q_N|, |Q_W|, |Q_S|, |Q_E| \rangle$ ▷ Step 2
    $\langle 1,1,1,1 \rangle : \mathbb{P}(s,T) = Case1(G_k,T,s)$;
    $\langle 2,0,2,0 \rangle \vee \langle 0,2,0,2 \rangle : \mathbb{P}(s,T) = Case2(G_k,T,s)$;
    $\langle 2,2,0,0 \rangle \vee \langle 0,2,2,0 \rangle \vee \langle 0,0,2,2 \rangle \vee \langle 2,0,0,2 \rangle : \mathbb{P}(s,T) = Case3(G_k,T,s)$;
    $\langle 2,1,1,0 \rangle \vee \langle 0,2,1,1 \rangle \vee \langle 1,0,2,1 \rangle \vee \langle 1,1,0,2 \rangle : \mathbb{P}(s,T) = Case4(G_k,T,s)$;
    $\langle 2,0,1,1 \rangle \vee \langle 1,2,0,1 \rangle \vee \langle 1,1,2,0 \rangle \vee \langle 0,1,1,2 \rangle : \mathbb{P}(s,T) = Case5(G_k,T,s)$;
    $\langle 2,1,0,1 \rangle \vee \langle 1,2,1,0 \rangle \vee \langle 0,1,2,1 \rangle \vee \langle 1,0,1,2 \rangle : \mathbb{P}(s,T) = Case6(G_k,T,s)$;
    $\langle 3,0,0,1 \rangle \vee \langle 1,3,0,0 \rangle \vee \langle 0,1,3,0 \rangle \vee \langle 0,0,1,3 \rangle : \mathbb{P}(s,T) = Case7(G_k,T,s)$;
    $\langle 3,1,0,0 \rangle \vee \langle 0,3,1,0 \rangle \vee \langle 0,0,3,1 \rangle \vee \langle 1,0,0,3 \rangle : \mathbb{P}(s,T) = Case8(G_k,T,s)$;
    $\langle 3,0,1,0 \rangle \vee \langle 0,3,0,1 \rangle \vee \langle 1,0,3,0 \rangle \vee \langle 0,1,0,3 \rangle : \mathbb{P}(s,T) = Case9(G_k,T,s)$;
    $\langle 4,0,0,0 \rangle \vee \langle 0,4,0,0 \rangle \vee \langle 0,0,4,0 \rangle \vee \langle 0,0,0,4 \rangle : \mathbb{P}(s,T) = Case10(G_k,T,s)$;
  end switch
  return $\mathbb{P}(s,T)$;
end procedure

3.1. Step 1: Case Determination

The Gaussian network G_k can be partitioned into four non-overlapped quadrants based on the source node's address. For any source node $s = (s_x, s_y)$, these quadrants are:

1. $Q_N = \{(x,y) \in G_k \mid (x \ge s_x) \wedge (y \ge s_y + 1)\}$ (The north quadrant)
2. $Q_W = \{(x,y) \in G_k \mid (x \le s_x - 1) \wedge (y \ge s_y)\}$ (The west quadrant)
3. $Q_S = \{(x,y) \in G_k \mid (x \le s_x) \wedge (y \le s_y - 1)\}$ (The south quadrant)

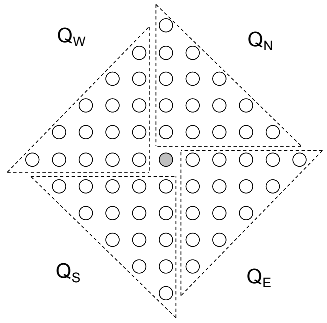


FIGURE 6: The Quadrants (G_5)

TABLE 1: All Cases

Case No.	Chosen Cases	Equivalent Cases
1	$\langle 1,1,1,1\rangle$	no equivalent case
2	$\langle 2,0,2,0\rangle$	$\langle 0,2,0,2\rangle$
3	$\langle 2,2,0,0\rangle$	$\langle 0,2,2,0\rangle$, $\langle 0,0,2,2\rangle$, $\langle 2,0,0,2\rangle$
4	$\langle 2,1,1,0\rangle$	$\langle 0,2,1,1\rangle$, $\langle 1,0,2,1\rangle$, $\langle 1,1,0,2\rangle$
5	$\langle 2,0,1,1\rangle$	$\langle 1,2,0,1\rangle$, $\langle 1,1,2,0\rangle$, $\langle 0,1,1,2\rangle$
6	$\langle 2,1,0,1\rangle$	$\langle 1,2,1,0\rangle$, $\langle 0,1,2,1\rangle$, $\langle 1,0,1,2\rangle$
7	$\langle 3,0,0,1\rangle$	$\langle 1,3,0,0\rangle$, $\langle 0,1,3,0\rangle$, $\langle 0,0,1,3\rangle$
8	$\langle 3,1,0,0\rangle$	$\langle 0,3,1,0\rangle$, $\langle 0,0,3,1\rangle$, $\langle 1,0,0,3\rangle$
9	$\langle 3,0,1,0\rangle$	$\langle 0,3,0,1\rangle$, $\langle 1,0,3,0\rangle$, $\langle 0,1,0,3\rangle$
10	$\langle 4,0,0,0\rangle$	$\langle 0,4,0,0\rangle$, $\langle 0,0,4,0\rangle$, $\langle 0,0,0,4\rangle$

4. $Q_E = \{(x,y) \in G_k \mid (x \geq s_x + 1) \wedge (y \leq s_y)\}$ (The east quadrant)

Each quadrant has exactly $k(k+1)/2$ nodes where k is the network diameter.

In case the source node s is $(0,0)$ as we have assumed in this work, the quadrants are (see Figure6):

1. $Q_N = \{(x,y) \in G_k \mid (x \geq 0) \wedge (y \geq 1)\}$
2. $Q_W = \{(x,y) \in G_k \mid (x \leq -1) \wedge (y \geq 0)\}$
3. $Q_S = \{(x,y) \in G_k \mid (x \leq 0) \wedge (y \leq -1)\}$
4. $Q_E = \{(x,y) \in G_k \mid (x \geq 1) \wedge (y \leq 0)\}$

For the Gaussian network G_5 as shown in Figure6, the number of nodes in each quadrants is equal to $5(5+1)/2 = 15$ where the diameter $k = 5$.

Based on this network partitioning, the algorithm determines the current case. Let $|Q_i| \in \{0,1,2,3,4\}$ be the number of destination nodes in the quadrant Q_i for $i = N, W, S, E$. Let the ordered set $\langle |Q_N|, |Q_W|, |Q_S|, |Q_E|\rangle$ represent the number of destination nodes in all quadrants such that $|Q_N| + |Q_W| + |Q_S| + |Q_E| = 4$. For example, $\langle 4,0,0,0\rangle$ means all destination nodes are in the north quadrant.

Since there are four destination nodes that are distributed over the four quadrants, there are exactly $\binom{4+4-1}{4} = 35$ possibilities of $\langle |Q_N|, |Q_W|, |Q_S|, |Q_E|\rangle$. The one-to-many NDP routing algorithm must construct all NDP $\mathbb{P}(s,T)$ for each one of these 35 possibilities. However, since G_k is vertex symmetric, the solution for $\langle x_1,x_2,x_3,x_4\rangle$ is equivalent to the solutions for $\langle x_4,x_1,x_2,x_3\rangle$,$\langle x_3,x_4,x_1,x_2\rangle$, and $\langle x_2,x_3,x_4,x_1\rangle$ (by rotation[1]) where $x_1,x_2,x_3,x_4 \in \{0,1,2,3,4\}$ and $\sum_{i=1}^{4} x_i = 4$. So in this work, we show the NDP $\mathbb{P}(s,T)$ for 10 cases. The solutions for these 10 cases are equivalent to the solutions for all 35 cases. Table 1 shows the chosen 10 cases and the equivalent cases; the total is 35 cases.

Based on the destination nodes' addresses, the algorithm evaluates $\langle |Q_N|, |Q_W|, |Q_S|, |Q_E|\rangle$ in the first step. In the second step, the algorithm constructs the NDP $\mathbb{P}(s,T)$.

3.2. Step 2: One-to-Many NDP Construction

In this step, the algorithm constructs four NDP from the source node to the destination nodes based on the case determined during the first step. In the following, we describe the NDP construction for each case of the 10 cases. Before that, we need the following definitions.

Definition 3.1. *In a Gaussian network G_k where k is the diameter, let the source node be $(0,0)$. Then, the north, west, south, and east node disjoint path starts with $(0,0) \to (0,1)$, $(0,0) \to (-1,0)$, $(0,0) \to (0,-1)$, and $(0,0) \to (1,0)$, respectively.*

Definition 3.2. *In a Gaussian network G_k where k is the diameter, let $t_j = (t_{j_x}, t_{j_y}) \in Q_i$ for $j = 1,2,3,4$ and $i = N,W,S,E$ be any destination node. Then, the destination node t_j is:*

- *the top destination node of Q_i if $t_{j_y} = \max\{t_{r_y} | t_r = (t_{r_x}, t_{r_y}) \in Q_i\}$,*
- *the bottom destination node of Q_i if $t_{j_y} = \min\{t_{r_y} | t_r = (t_{r_x}, t_{r_y}) \in Q_i\}$,*
- *the left destination node of Q_i if $t_{j_x} = \min\{t_{r_x} | t_r = (t_{r_x}, t_{r_y}) \in Q_i\}$,*
- *the right destination node of Q_i if $t_{j_x} = \max\{t_{r_x} | t_r = (t_{r_x}, t_{r_y}) \in Q_i\}$,*
- *the max-weight destination node of Q_i if $W(t_j) = \max\{W(t_r) | t_r = (t_{r_x}, t_{r_y}) \in Q_i\}$, and/or*
- *the min-weight destination node of Q_i if $W(t_j) = \min\{W(t_r) | t_r = (t_{r_x}, t_{r_y}) \in Q_i\}$.*

Note that the top, bottom, left, right, max-weight, or min-weight destination node as defined in the Definition 3.2 is not necessarily unique. So, we say, for example, top/left of Q_i to uniquely specify a destination

[1] multiplying all nodes

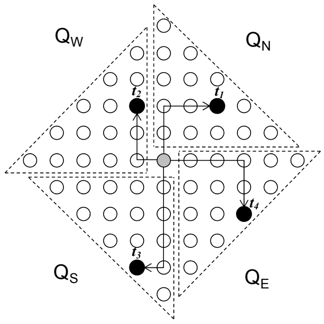


FIGURE 7: Example of Case 1 (G_5)

node in case the top destination node is not unique by choosing the most left destination node among those top destination nodes.

Now, we explain how to construct the NDP for each case.

3.2.1. *Case 1* $\langle 1, 1, 1, 1\rangle$

In this case, each quadrant has exactly one destination node and this is the most simple case. The node disjoint path to the destination node in the north, west, south, and east quadrant is connected along the north, west, south, and east path, respectively. These NDP are formally given in the proof of the following lemma and Figure7 shows an example.

LEMMA 3.1. *In the Gaussian network G_k where k is the network diameter, let the source node be $s = (0, 0)$ and the set of destination nodes be $T = \{t_j = (t_{j_x}, t_{j_y}) | 1 \le j \le 4\}$ such that the case is $\langle 1, 1, 1, 1\rangle$. Then, there exist NDP $\mathbb{P}(s, T)$ such that the sum of the lengths of the paths in $\mathbb{P}(s, T)$ is exactly equal to*

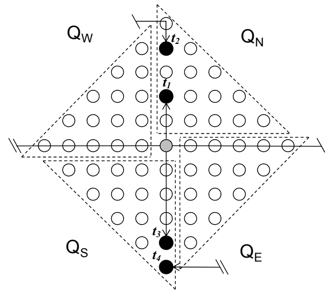
$$|\mathbb{P}(s, T)| = L(T)$$

Proof. Assume, without loss of generality, that $t_1 \in Q_N$, $t_2 \in Q_W$, $t_3 \in Q_S$, $t_4 \in Q_E$. Then, the NDP $\mathbb{P}(s, T)$ are:

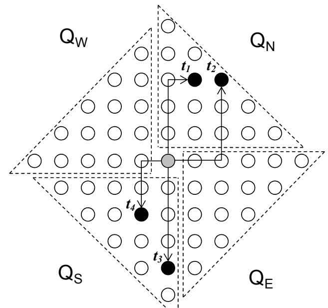
1. $P(s, t_1)$ is $(0, 0) \to (0, 1) \to (0, 2) \to \cdots \to (0, t_{1_y}) \to (1, t_{1_y}) \to (2, t_{1_y}) \to \cdots \to (t_{1_x}, t_{1_y})$. Its length is equal to $W(t_1)$.
2. $P(s, t_2)$ is $(0, 0) \to (-1, 0) \to (-2, 0) \to \cdots \to (t_{2_x}, 0) \to (t_{2_x}, 1) \to (t_{2_x}, 2) \to \cdots \to (t_{2_x}, t_{2_y})$. Its length is equal to $W(t_2)$.
3. $P(s, t_3)$ is $(0, 0) \to (0, -1) \to (0, -2) \to \cdots \to (0, t_{3_y}) \to (-1, t_{3_y}) \to (-2, t_{3_y}) \to \cdots \to (t_{3_x}, t_{3_y})$. Its length is equal to $W(t_3)$.
4. $P(s, t_4)$ is $(0, 0) \to (1, 0) \to (2, 0) \to \cdots \to (t_{4_x}, 0) \to (t_{4_x}, -1) \to (t_{4_x}, -2) \to \cdots \to (t_{4_x}, t_{4_y})$. Its length is equal to $W(t_4)$.

It follows that $|\mathbb{P}(s, T)| = \sum_{j=1}^{4} |P(s, t_j)| = \sum_{j=1}^{4} W(t_j) = L(T)$.

□



(a) Case 2.1



(b) Case 2.2

FIGURE 8: Example of Case 2 (G_5)

3.2.2. *Case 2* $\langle 2, 0, 2, 0\rangle$

In this case, the north quadrant Q_N and the south quadrant Q_S have two destination nodes each. The NDP to the destination nodes in the north quadrant Q_N are connected along the north and east paths; and the NDP to the destination nodes in the south quadrant Q_S are connected along the west and south paths. These NDP are formally given in the proof of the following lemma and Figure8 shows an example.

LEMMA 3.2. *In the Gaussian network G_k where k is the network diameter, let the source node be $s = (0, 0)$ and the set of destination nodes be $T = \{t_j = (t_{j_x}, t_{j_y}) | 1 \le j \le 4\}$ such that the case is $\langle 2, 0, 2, 0\rangle$. Then, there exist NDP $\mathbb{P}(s, T)$ such that the sum of the lengths of the paths in $\mathbb{P}(s, T)$ is*

$$L(T) \le |\mathbb{P}(s, T)| \le L(T) + (4k - 6)$$

Proof. Let $t_1, t_2 \in Q_N$ and $t_3, t_4 \in Q_S$. Case 2 can be divided into the following four subcases:

Case 2.1 $t_{1_x} = t_{2_x} = t_{3_x} = t_{4_x} = 0$:

In this case, the NDP to the bottom destination node of Q_N and the top destination node of Q_S are connected along the y-axis. The NDP to the top destination node of Q_N and the bottom destination node of Q_S are respectively connected along the positive x-axis and negative x-axis with wraparound links.

Assume, without loss of generality, that:

1. t_1 and t_2 are respectively the bottom and top destination nodes of Q_N, and
2. t_3 and t_4 are respectively the top and bottom destination nodes of Q_S.

Then, the NDP $\mathbb{P}(s,T)$ are (see Figure8a):

1. $P(s,t_1)$ is $(0,0) \rightarrow (0,1) \rightarrow \cdots \rightarrow (0,t_{1_y})$. Its length is equal $W(t_1)$.
2. $P(s,t_2)$ is $(0,0) \rightarrow (1,0) \rightarrow \cdots \rightarrow (k,0) \rightarrow (k,0)^E = (0,k) \rightarrow (0,k-1) \rightarrow \cdots \rightarrow (0,t_{2_y})$. Since the minimum and maximum length of this path occur respectively when $t_2 = (0,k)$ and $t_2 = (0,2)$, the length of $P(s,t_2)$ is greater than or equal $R((0,k)) = W((0,k)) + 2(k - W((0,k))) + 1 = W((0,k)) + 1$ and less than or equal $R((0,2)) = W((0,2)) + 2(k - W((0,2))) + 1 = W((0,2)) + (2k-3)$.
3. $P(s,t_3)$ is $(0,0) \rightarrow (0,-1) \rightarrow \cdots \rightarrow (0,t_{3_y})$. Its length is equal to $W(t_3)$.
4. $P(s,t_4)$ is $(0,0) \rightarrow (-1,0) \rightarrow \cdots \rightarrow (-k,0) \rightarrow (-k,0)^W = (0,-k) \rightarrow (0,-k+1) \rightarrow \cdots \rightarrow (0,t_{4_y})$. Similar to $P(s,t_2)$, the path length of $P(s,t_4)$ is $W((0,-k)) + 1 \leq |P(s,t_4)| \leq W((0,-2)) + (2k-3)$.

It follows that $\sum_{j=1}^{4} |P(s,t_j)| = L(T) + 2 \leq |\mathbb{P}(s,T)| \leq \sum_{j=1}^{4} |P(s,t_j)| = L(T) + (4k-6)$.

Case 2.2 ($t_{1_x} \neq 0$ or $t_{2_x} \neq 0$)and($t_{3_x} \neq 0$ or $t_{4_x} \neq 0$):

In this case, there exists at least one destination node in each quadrant such that its x value is not equal to zero. The node disjoint path to this destination node in Q_N and Q_S is connected along the east and west paths respectively. The other destination node in Q_N and Q_S is reached using the north and south paths respectively.

Assume, without loss of generality, that:

1. $t_{1_x} = 0$ or t_1 is the top/left destination node of Q_N,
2. t_2 is the bottom/right destination node of Q_N,
3. $t_{3_x} = 0$ or t_3 is the bottom/right destination node of Q_S, and
4. t_4 is the top/left destination node of Q_S.

Then, the NDP $\mathbb{P}(s,T)$ are (see Figure8b):

1. $P(s,t_1)$ is same as the north path in Case 1.
2. $P(s,t_2)$ is $(0,0) \rightarrow (1,0) \rightarrow (2,0) \rightarrow \cdots \rightarrow (t_{2_x},0) \rightarrow (t_{2_x},1) \rightarrow (t_{2_x},2) \rightarrow \cdots \rightarrow (t_{2_x},t_{2_y})$. Its length is equal to $W(t_2)$.
3. $P(s,t_3)$ is same as the south path in Case 1.
4. $P(s,t_4)$ is $(0,0) \rightarrow (-1,0) \rightarrow (-2,0) \rightarrow \cdots \rightarrow (t_{4_x},0) \rightarrow (t_{4_x},-1) \rightarrow (t_{4_x},-2) \rightarrow \cdots \rightarrow (t_{4_x},t_{4_y})$. Its length is equal to $W(t_4)$.

TABLE 2: All subcases of Case 2

Case No.	Lower Bound	Upper Bound
2.1	$L(T)+2$	$L(T)+(4k-6)$
2.2	$L(T)$	$L(T)$
2.3	$L(T)+1$	$L(T)+(2k-3)$
2.4	$L(T)+1$	$L(T)+(2k-3)$

It follows that $|\mathbb{P}(s,T)| = \sum_{j=1}^{4} |P(s,t_j)| = \sum_{j=1}^{4} W(t_j) = L(T)$.

Case 2.3 ($t_{1_x} = t_{2_x} = 0$)and($t_{3_x} \neq 0$ or $t_{4_x} \neq 0$):

In this case, the NDP to the destination nodes in Q_N (i.e. t_1 and t_2) are exactly same as the the north and east paths in Case 2.1. Moreover, the NDP to the destination nodes in Q_S (i.e. t_3 and t_4) are exactly same as the the west and south paths in Case 2.2. Clearly, there exist NDP $\mathbb{P}(s,T)$ such that $L(T)+1 \leq |\mathbb{P}(s,T)| \leq L(T) + (2k-3)$.

Case 2.4 ($t_{1_x} \neq 0$ or $t_{2_x} \neq 0$)and($t_{3_x} = t_{4_x} = 0$):

Similar to Case 2.3 except that the north and east paths in this case are same as the the north and east paths in Case 2.2; and the west and south paths are same as the the west and south paths in Case 2.1. Clearly, there exist NDP $\mathbb{P}(s,T)$ such that $L(T)+1 \leq |\mathbb{P}(s,T)| \leq L(T) + (2k-3)$.

Table 2 shows the upper and lower bounds of all subcases of Case 2. Clearly, the minimum lower bound and the maximum upper bound occur when the cases are Case 2.2 and Case 2.1 respectively. It follows that $L(T) \leq |\mathbb{P}(s,T)| \leq L(T) + (4k-6)$. □

In the upcoming cases, Case 2 is used to reach two destination nodes in Q_N as long as none of the following nodes is used: $(1,0),(2,0),\ldots,(k-1,0),(k,0)$. Similarly, Case 2 is used to reach two destination nodes in Q_S as long as none of the following nodes is used: $(-1,0),(-2,0),\ldots,(-k+1,0),(-k,0)$.

3.2.3. Case 3: $\langle 2,2,0,0 \rangle$

In this case, the north and west quadrants have two destination nodes each. The node disjoint path to the min-weight/right destination node of Q_W is connected along the west path; and the node disjoint path to the max-weight/left destination node of Q_W is connected along the south path. By the Gaussian network connectivity, the south path does not use any node of $(1,0),(2,0),\ldots,(k-1,0),(k,0)$ (Figure9 shows all border nodes (dashed) that can be used by the south path). So, Case 2 is used to reach the destination nodes

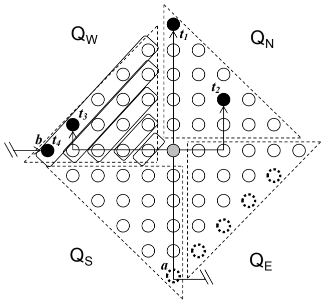


FIGURE 9: Example of Case 3 (G_5)

in Q_N using the north and east paths. These NDP are formally given in the proof of the following lemma and Figure9 shows an example.

LEMMA 3.3. *In the Gaussian network G_k where k is the network diameter, let the source node be $s = (0,0)$ and the set of destination nodes be $T = \{t_j = (t_{j_x}, t_{j_y}) | 1 \le j \le 4\}$ such that the case is $\langle 2,2,0,0\rangle$. Then, there exist NDP $\mathbb{P}(s,T)$ such that the sum of the lengths of the paths in $\mathbb{P}(s,T)$ is*

$$L(T) + 1 \le |\mathbb{P}(s,T)| \le L(T) + (4k-6)$$

Proof. Let $t_1, t_2 \in Q_N$ and $t_3, t_4 \in Q_W$. Assume, without loss of generality, that:

1. t_3 is the min-weight/right destination node of Q_W, and
2. t_4 is the max-weight/left destination node of Q_W.

Then, the NDP $\mathbb{P}(s,T)$ are:

1. $P(s,t_1)$ and $P(s,t_2)$ are obtained by applying Case 2. It follows that $(W(t_1)+W(t_2)) \le (|P(s,t_1)| + |P(s,t_2)|) \le (W(t_1)+W(t_2)+2k-3)$.
2. $P(s,t_3)$ is same as the west path in Case 1. Its length is equal to $W(t_3)$.
3. Let $b = (b_x, t_{4_y}) \in Q_W$ be a border node and $a = (a_x, a_y) = b^W \in Q_E$ be another border node. Then, $P(s,t_4)$ is $(0,0) \to (0,-1) \to (0,-2) \to \cdots \to (0,a_y) \to (1,a_y) \to \cdots \to (a_x,a_y) \to (b_x,t_{4_y}) \to (b_x+1,t_{4_y}) \to \cdots \to (t_{4_x},t_{4_y})$. Its length is at most equal to $W(t_4)+(2k-3)$ which occurs when $W(t_4)=2$. Also, this length is at least equal to $W(t_4)+1$ which occurs when $W(t_4)=k$.

It follows that $L(T)+1 \le |\mathbb{P}(s,T)| \le L(T)+(4k-6)$. □

3.2.4. Case 4: $\langle 2,1,1,0\rangle$

In this case, there exist two destination nodes in Q_N, one destination node in Q_W, and one destination node in Q_S. Since the east quadrant Q_E has no destination nodes, none of the nodes $(1,0),(2,0),\ldots,(k-1,0),(k,0)$ is used. So, Case 2 can be used to reach the two destination nodes in Q_N using the north and east paths. These NDP are formally given in the proof of the following lemma and Figure10 shows an example.

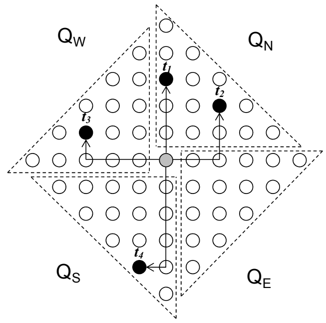


FIGURE 10: Example of Case 4 (G_5)

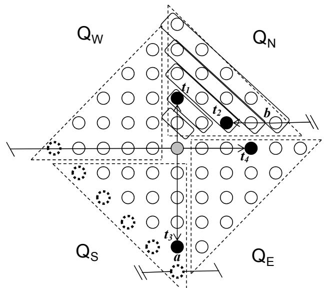


FIGURE 11: Example of Case 5 (G_5)

LEMMA 3.4. *In the Gaussian network G_k where k is the network diameter, let the source node be $s = (0,0)$ and the set of destination nodes be $T = \{t_j = (t_{j_x}, t_{j_y}) | 1 \le j \le 4\}$ such that the case is $\langle 2,1,1,0\rangle$. Then, there exist NDP $\mathbb{P}(s,T)$ such that the sum of the lengths of the paths in $\mathbb{P}(s,T)$ is*

$$L(T) \le |\mathbb{P}(s,T)| \le L(T) + (2k-3)$$

Proof. Let $t_1, t_2 \in Q_N$, $t_3 \in Q_W$, and $t_4 \in Q_S$. The NDP $\mathbb{P}(s,T)$ are:

1. $P(s,t_1)$ and $P(s,t_2)$ are obtained by applying Case 2. It follows that $(W(t_1)+W(t_2)) \le (|P(s,t_1)| + |P(s,t_2)|) \le (W(t_1)+W(t_2)+2k-3)$.
2. $P(s,t_3)$ and $P(s,t_4)$ are same as the west and south paths in Case 1 respectively. The sum of their lengths is equal to $|P(s,t_3)| + |P(s,t_4)| = W(t_3)+W(t_4)$.

It follows that $L(T) \le |\mathbb{P}(s,T)| \le L(T)+(2k-3)$. □

3.2.5. Case 5: $\langle 2,0,1,1\rangle$

In this case, there exist two destination nodes in Q_N, one destination node in Q_S, and one destination node in Q_E. Since the east quadrant Q_E has one destination node, one of the nods in $(1,0),(2,0),\ldots,(k-1,0),(k,0)$

can be used by this destination node. So, Case 2 cannot be used to reach the two destination nodes in Q_N using the north and east paths. However in the west quadrant Q_E, none of the nodes in $(-1,0),(-2,0),\ldots,(-(k-1),0),(-k,0)$ is used because the west quadrant has no destination nodes. Thus, Case 2 can be used to reach two destination nodes in the south quadrant Q_S using the west and south paths. Since there is only one destination node in the south quadrant Q_S, the algorithm connects one of the destination nodes in the north quadrant Q_N with a border node in the south quadrant Q_S and then uses Case 2. These NDP are formally given in the proof of the following lemma and Figure11 shows an example.

Lemma 3.5. *In the Gaussian network G_k where k is the network diameter, let the source node be $s=(0,0)$ and the set of destination nodes be $T=\{t_j=(t_{j_x},t_{j_y})|1\le j\le 4\}$ such that the case is $\langle 2,0,1,1\rangle$. Then, there exist NDP $\mathbb{P}(s,T)$ such that the sum of the lengths of the paths in $\mathbb{P}(s,T)$ is*

$$L(T)+1\le|\mathbb{P}(s,T)|\le L(T)+(2k-2)$$

Proof. Let $t_1,t_2\in Q_N$, $t_3\in Q_S$, and $t_4\in Q_E$. The algorithm performs the following steps to construct the NDP:

1. Reach the destination node t_4 in Q_E using the east path.

2. Reach the min-weight/left destination node of Q_N, say t_1, using the north path.

3. Connect the max-weight/right destination node of Q_N, say t_2, with the border node $b=(b_x,t_{2_y})\in Q_N$ vertically using the path $P(b,t_2)$ as $(b_x,t_{2_y})\to(b_x-1,t_{2_y})\to\cdots\to(t_{2_x},t_{2_y})$.

4. Let $a=(a_x,a_y)$ be either the east (b^E) or north (b^N) neighbour of node b depending on the location of the destination node t_3 in Q_S as follows (the dashed nodes in Figure11 are all possibilities of node a):

$$a=\begin{cases} b^N & \text{if} \quad b^E=t_3 \\ b^E & \text{if} \quad b^E\neq t_3 \end{cases}$$

   Note that node a can be either in the west quadrant Q_W (if $t_2=(0,k)$ and $t_3=(-(k-1),-1)$) or the south quadrant Q_S (otherwise).

5. If node a is in the west quadrant Q_W, reach node a and the destination node t_3 using the west and south paths respectively.

6. If node a is in the south quadrant Q_S, use Case 2 to reach the node a and destination node t_3 using the west and south paths.

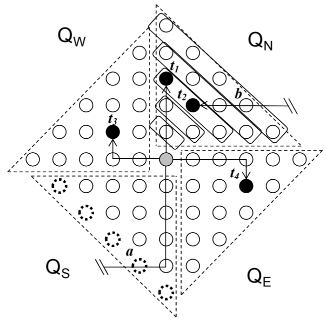


FIGURE 12: Example of Case 6 (G_5)

Step four is always possible because by the network connectivity each border node b in Q_N is connected with two nodes (b^N and b^E) using the wraparound links. One of these two nodes must be available to use because there exists only one destination node in Q_S. In case a is in Q_W, step four is still valid because Q_W has no destination node.

The NDP $\mathbb{P}(s,T)$ are:

1. $P(s,t_1)$ and $P(s,t_4)$ are same as the north and east path in Case 1 respectively. The sum of their lengths is equal to $|P(s,t_1)|+|P(s,t_4)|=W(t_1)+W(t_4)$.

2. $P(s,t_2)$ is divided into $P(s,a)$ and $P(b,t_2)$ where a and b are neighbors as explained above. If $a\in Q_W$, $P(s,a)$ is same as the west path in Case 1. If $a\in Q_S$, $P(s,a)$ is obtained by applying Case 2. $P(b,t_2)$ is $(b_x,t_{2_y})\to(b_x-1,t_{2_y})\to\cdots\to(t_{2_x},t_{2_y})$. The length of $P(s,t_2)$ is at most equal to $W(t_2)+(2k-2)$ which occurs when $W(t_2)=2$ and $a=(0,-k)$. Moreover, the length of $P(s,t_2)$ is at least equal to $W(t_2)+1$ which occurs when $W(t_2)=k$ and the following is not true: $a_x=t_{3_x}=0$.

3. If $a\in Q_W$, $P(s,t_3)$ is same as the south path in Case 1. If $a\in Q_S$, $P(s,t_3)$ is obtained by applying Case 2. In both cases, the length of $P(s,t_3)$ is equal to $W(t_3)$.

It follows that $L(T)+1\le|\mathbb{P}(s,T)|\le L(T)+(2k-2)$. □

3.2.6. Case 6: $\langle 2,1,0,1\rangle$

In this case, there exist two destination nodes in Q_N, one destination node in Q_W, and one destination node in Q_E. The NDP to the destination nodes in Q_W and Q_E are respectively connected along the west and east paths. The node disjoint path to the min-weight/left destination node of Q_N is connected along the north path while the other destination node in Q_N is reached using the south path. These NDP are formally given in the proof of the following lemma and Figure12 shows an example.

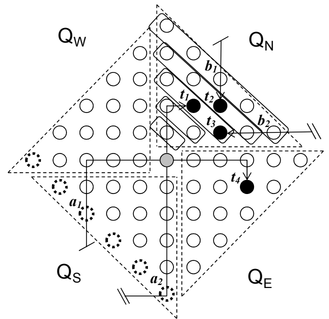


FIGURE 13: Example of Case 7 (G_5)

Lemma 3.6. *In the Gaussian network G_k where k is the network diameter, let the source node be $s = (0,0)$ and the set of destination nodes be $T = \{t_j = (t_{j_x}, t_{j_y}) | 1 \leq j \leq 4\}$ such that the case is $\langle 2,1,0,1 \rangle$. Then, there exist NDP $\mathbb{P}(s,T)$ such that the sum of the lengths of the paths in $\mathbb{P}(s,T)$ is*

$$L(T) + 1 \leq |\mathbb{P}(s,T)| \leq L(T) + (2k-3)$$

Proof. Let $t_1, t_2 \in Q_N$, $t_3 \in Q_W$, and $t_4 \in Q_E$. Assume, without loss of generality, that:

1. t_1 is the min-weight/left destination node of Q_N, and
2. t_2 is the max-weight/right destination node of Q_N.

Formally, the NDP $\mathbb{P}(s,T)$ are:

1. $P(s,t_1)$, $P(s,t_3)$, and $P(s,t_4)$ are same as the north, west, and east paths in Case 1 respectively. The sum of their lengths is equal to $W(t_1) + W(t_3) + W(t_4)$.
2. Let $b = (b_x, t_{2_y}) \in Q_N$ be a border node and $a = (a_x, a_y) = b^E \in Q_S$ be another border node. Then, $P(s,t_2)$ is $(0,0) \to (0,-1) \to (0,-2) \to \cdots \to (0,a_y) \to (-1,a_y) \to \cdots \to (a_x,a_y) \to (b_x,t_{2_y}) \to (b_x-1,t_{2_y}) \to \cdots \to (t_{2_x},t_{2_y})$. This path is valid because by the network connectivity all possibilities of node a (the dashed nodes in Figure12) are in Q_S which has no destination nodes. The length of $P(s,t_2)$ is at most equal to $W(t_2) + (2k-3)$ which occurs when $W(t_2) = 2$. Also, this length is at least equal to $W(t_2)+1$ which occurs when $W(t_2) = k$.

It follows that $L(T) + 1 \leq |\mathbb{P}(s,T)| \leq L(T) + (2k-3)$. □

3.2.7. Case 7: $\langle 3,0,0,1 \rangle$

In this case, there exist three destination nodes in Q_N and one destination node in Q_E. The node disjoint path to the destination node in Q_E is connected along the east path.

Since Q_W has no destination nodes, none of the nodes in $(-1,0), (-2,0), \ldots, (-(k-1),0), (-k,0)$ is used. So, Case 5 can be used to reach two destination nodes in Q_S (or one of them in Q_S and the other in Q_W) using the south and west paths. To use Case 5, the algorithm connects two destination nodes from Q_N with Q_S (or Q_S and Q_W) using NDP. These NDP are formally given in the proof of the following lemma and Figure13 shows an example.

Lemma 3.7. *In the Gaussian network G_k where k is the network diameter, let the source node be $s = (0,0)$ and the set of destination nodes be $T = \{t_j = (t_{j_x}, t_{j_y}) | 1 \leq j \leq 4\}$ such that the case is $\langle 3,0,0,1 \rangle$. Then, there exist NDP $\mathbb{P}(s,T)$ such that the sum of the lengths of the paths in $\mathbb{P}(s,T)$ is*

$$L(T) + 2 \leq |\mathbb{P}(s,T)| \leq L(T) + (4k-6)$$

Proof. Let $t_1, t_2, t_3 \in Q_N$, and $t_4 \in Q_E$. The algorithm performs the following steps to construct the NDP:

1. Reach the destination node t_4 using the east path.
2. Reach the the min-weight/left destination node of Q_N, say t_1, using the north path.
3. After the previous step, the remaining destination nodes in Q_N are t_2 and t_3. Among these two destination nodes, connect the top/left destination node of Q_N, say t_2, with the border node $b_1 = (t_{2_x}, b_{1_y}) \in Q_N$ vertically using the path $P(b_1,t_2)$ as $(t_{2_x}, b_{1_y}) \to (t_{2_x}, b_{1_y}-1) \to \cdots \to (t_{2_x}, t_{2_y})$.
4. Among t_2 and t_3, connect the bottom/right destination node of Q_N, say t_3, with the border node $b_2 = (b_{2_x}, t_{3_y}) \in Q_N$ horizontally using the path $P(b_2,t_3)$ as $(b_{2_x}, t_{3_y}) \to (b_{2_x}-1, t_{3_y}) \to \cdots \to (t_{3_x}, t_{3_y})$.
5. Apply Case 5 to connect the source node with the north neighbor of b_1, denoted by $a_1 = b_1^N$, and the east neighbor of b_2, denoted by $a_2 = b_2^E$, using the west and south paths. Figure13 shows all possibilities of a_1 and a_2 (the dashed nodes).

Constructing the path $P(b_1,t_2)$ vertically and the path $P(b_2,t_3)$ horizontally is important to maintain the disjointness condition for two reasons: 1) in Q_N, $P(b_1,t_2)$ and $P(b_2,t_3)$ are always node disjoint regardless of the locations of t_2 and t_3, and 2) in Q_S and Q_W, the north neighbor of b_1 and the east neighbor of b_2 are always different nodes. The NDP $\mathbb{P}(s,T)$ are:

1. $P(s,t_1)$ and $P(s,t_4)$ are same as the north and east paths in Case 1 respectively. The sum of their lengths is equal to $W(t_1) + W(t_4)$.
2. $P(s,t_2)$ is divided into $P(s,a_1)$ and $P(b_1,t_2)$ where a_1 and b_1 are neighbors as explained above. $P(b_1,t_2)$ is $(t_{2_x}, b_{1_y}) \to (t_{2_x}, b_{1_y}-1) \to \cdots \to (t_{2_x}, t_{2_y})$. $P(s,a_1)$ is obtained by applying Case

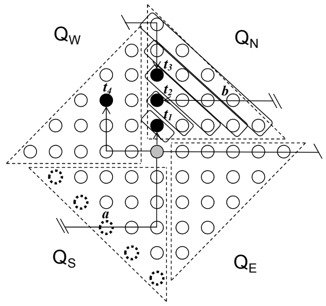


FIGURE 14: Example of Case 8 (G_5)

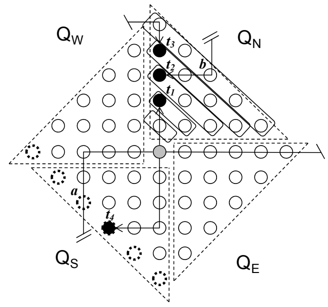


FIGURE 15: Example of Case 9 (G_5)

5. The length of $P(s,t_2)$ is at most equal to $W(t_2)+(2k-3)$ which occurs when $W(t_2)=2$. Also, this length is at least equal to $W(t_2)+1$ which occurs when $W(t_2)=k$.

3. $P(s,t_3)$ is divided into $P(s,a_2)$ and $P(b_2,t_3)$ where a_2 and b_2 are neighbors as explained above. $P(b_2,t_3)$ is $(b_{2_x},t_{3_y}) \rightarrow (b_{2_x}-1,t_{3_y}) \rightarrow \cdots \rightarrow (t_{3_x},t_{3_y})$. $P(s,a_2)$ is obtained by applying Case 5. The length of $P(s,t_3)$ is at most equal to $W(t_3)+(2k-3)$ when $W(t_3)=2$. Also, this length is at least equal to $W(t_3)+1$ which occurs when $W(t_3)=k$.

It follows that $L(T)+2 \leq |\mathbb{P}(s,T)| \leq L(T)+(4k-6)$. □

3.2.8. *Case 8:* $\langle 3,1,0,0\rangle$

In this case, there exist three destination nodes in Q_N and one destination node in Q_W. The node disjoint path to the destination node in Q_W is connected along the west path.

Since none of the nodes in $(1,0),(2,0),\ldots,(k-1,0),(k,0)$ is used, the algorithm uses Case 2 to reach two destination nodes in Q_N using the north and east paths. The remaining destination node in Q_N is reached using the south path with a wraparound link. These NDP are formally given in the proof of the following lemma and Figure14 shows an example.

Lemma 3.8. *In the Gaussian network G_k where k is the network diameter, let the source node be $s=(0,0)$ and the set of destination nodes be $T=\{t_j=(t_{j_x},t_{j_y}) | 1\leq j\leq 4\}$ such that the case is $\langle 3,1,0,0\rangle$. Then, there exist NDP $\mathbb{P}(s,T)$ such that the sum of the lengths of the paths in $\mathbb{P}(s,T)$ is*

$$L(T)+1 \leq |\mathbb{P}(s,T)| \leq L(T)+(4k-6)$$

Proof. Let $t_1,t_2,t_3 \in Q_N$, and $t_4 \in Q_W$. The algorithm performs the following steps to construct the NDP:

1. Reach the destination node t_4 using the west path.
2. Let the destination node t_2 be $t_{1_y} < t_{2_y} < t_{3_y}$ if $t_{1_x}=t_{2_x}=t_{3_x}=0$. Otherwise, let the destination node t_2 be the max-weight/right destination node of Q_N. Then, reach the destination node t_2 using the south path with a wraparound link such that the portion of this path in Q_N is horizontal.
3. After the previous step, the remaining destination nodes in Q_N are t_1 and t_3. Apply Case 2 to reach them using the east and north paths.

In step 2, constructing the portion of the path to t_2 horizontally in Q_N is important to construct a node disjoint path because in this way all possibilities of the border node in Q_S (that this path goes through) are in Q_S which has no destination node. Figure14 shows all possibilities of the border node in Q_S (the dashed nodes) . The NDP $\mathbb{P}(s,T)$are:

1. $P(s,t_1)$ and $P(s,t_3)$ are obtained by applying Case 2. The sum of their lengths is $W(t_1)+W(t_3) \leq |P(s,t_1)|+|P(s,t_3)| \leq W(t_1)+W(t_3)+(2k-3)$.
2. $P(s,t_2)$ is same as the south path in Case 6. Its length is $W(t_2)+1 \leq |P(s,t_2)| \leq W(t_2)+(2k-3)$.
3. $P(s,t_4)$ is same as the west path in Case 1. Its length is equal to $W(t_4)$.

It follows that $L(T)+1 \leq |\mathbb{P}(s,T)| \leq L(T)+(4k-6)$. □

3.2.9. *Case 9:* $\langle 3,0,1,0\rangle$

In this case, there exist three destination nodes in Q_N and one destination node in Q_S. Since none of the nodes in $(1,0),(2,0),\ldots,(k-1,0),(k,0)$ is used, Case 2 can be used to reach two destination nodes in Q_N using the north and east paths. Moreover, none of the nodes in $(-1,0),(-2,0),\ldots,(-(k-1),0),(-k,0)$ is used. So, Case 5 can be used to reach the destination node in Q_S and a border node (in either Q_S or Q_W) using the west and south paths such that this border node is connected to one of the destination nodes in Q_N. These NDP are formally given in the proof of the following lemma and Figure15 shows an example.

Lemma 3.9. *In the Gaussian network G_k where k is the network diameter, let the source node be $s=$*

$(0,0)$ *and the set of destination nodes be* $T = \{t_j = (t_{j_x}, t_{j_y}) | 1 \le j \le 4\}$ *such that the case is* $\langle 3,0,1,0 \rangle$. *Then, there exist NDP* $\mathbb{P}(s,T)$ *such that the sum of the lengths of the paths in* $\mathbb{P}(s,T)$ *is*

$$L(T) + 1 \le |\mathbb{P}(s,T)| \le L(T) + (4k-5)$$

Proof. Let $t_1, t_2, t_3 \in Q_N$, and $t_4 \in Q_S$. The algorithm preforms the following steps to construct the NDP $\mathbb{P}(s,T)$:

1. Let the destination node t_2 be $t_{1_y} < t_{2_y} < t_{3_y}$ if $t_{1_x} = t_{2_x} = t_{3_x} = 0$. Otherwise, let the destination node t_2 be the max-weight/right destination node of Q_N. Then, connect t_2 with the border node $b = (b_x, t_{2_y})$ horizontally using the path $P(b, t_2)$ as $(b_x, t_{2_y}) \rightarrow (b_x - 1, t_{2_y}) \rightarrow \cdots \rightarrow (t_{2_x}, t_{2_y})$.

2. Let $a = (a_x, a_y)$ be either the east (b^E) or north (b^N) neighbour of node b depending on the location of the destination node t_4 as follows:

$$a = \begin{cases} b^N & \text{if} \quad b^E = t_4 \\ b^E & \text{if} \quad b^E \neq t_4 \end{cases}$$

   Figure15 shows all possibilities of node a (the dashed nodes). Note that node a can be either in the west quadrant Q_W (if $t_2 = (0,k)$ and $t_4 = (-(k-1),-1)$) or the south quadrant Q_S (otherwise). After specifying node a, apply Case 5 to reach a and t_4 using the west and south paths. The sum of the lengths of $P(s,t_2)$ and $P(s,t_4)$ is $W(t_2) + W(t_4) + 1 \le |P(s,t_2)| + |P(s,t_4)| \le W(t_2) + W(t_4) + (2k-2)$.

3. Apply Case 2 to reach t_1 and t_3 using the north and east paths. The sum of the lengths of $P(s,t_1)$ and $P(s,t_3)$ is $W(t_1) + W(t_3) \le |P(s,t_1)| + |P(s,t_3)| \le W(t_1) + W(t_3) + (2k-3)$.

It follows that $L(T) + 1 \le |\mathbb{P}(s,T)| \le L(T) + (4k-5)$. □

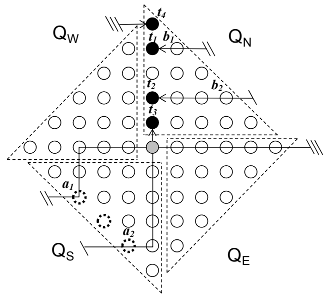
3.2.10. Case 10: $\langle 4,0,0,0 \rangle$

In this case, all four destination nodes are in Q_N and this is the most sophisticated case. The basic idea of constructing the NDP for this case is as follows:

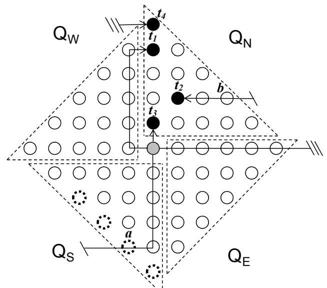
1. The NDP to two of the destination nodes in Q_N are connected along the north and east paths.

2. The remaining two destination nodes (sometimes one) are horizontally or vertically connected with Q_S and these are connected along the west and south paths.

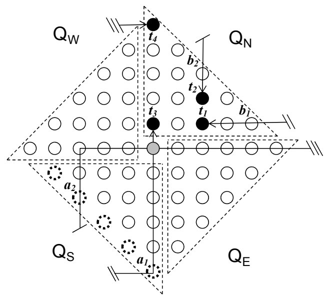
These NDP are formally given in the proof of the following lemma and Figure16 and Figure17 show some examples.



(a) Case 10.1



(b) Case 10.2



(c) Case 10.3

FIGURE 16: Examples of Case 10 (G_5)

LEMMA 3.10. *In the Gaussian network* G_k *where* k *is the network diameter, let the source node be* $s = (0,0)$ *and the set of destination nodes be* $T = \{t_j = (t_{j_x}, t_{j_y}) | 1 \le j \le 4\}$ *such that the case is* $\langle 4,0,0,0 \rangle$. *Then, there exist NDP* $\mathbb{P}(s,T)$ *such that the sum of the lengths of the paths in* $\mathbb{P}(s,T)$ *is*

$$L(T) + 2 \le |\mathbb{P}(s,T)| \le L(T) + (6k-11)$$

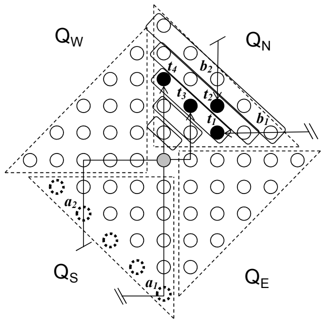
Proof. Here, $t_1, t_2, t_3, t_4 \in Q_N$. To show the construction of the NDP precisely, we divide this case into the following five subcases:

1. The x values of all destination nodes are equal to zero.

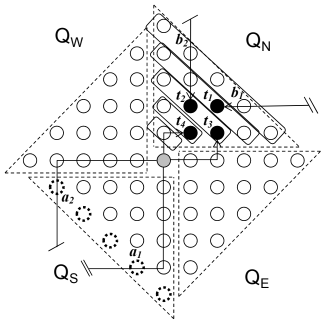
2. The x values of exactly three destination nodes are equal to zero.

3. The x values of exactly two destination nodes are equal to zero.

4. The x value of exactly one destination node is equal to zero.



(a) Case 10.4



(b) Case 10.5

FIGURE 17: Examples of Case 10 (G_5)

5. The x value of none of the destination node is equal to zero.

In the following, the algorithm constructs the NDP for each subcases.

Case 10.1 Four destination nodes have $x = 0$:

In this case, $t_{1_x} = t_{2_x} = t_{3_x} = t_{4_x} = 0$. Assume, without loss of generality, that $t_{3_y} < t_{2_y} < t_{1_y} < t_{4_y}$. In other words, t_1, t_2, t_3, and t_4 are the 2nd top, 2nd bottom, 1st bottom, and 1st top destination node of Q_N respectively (see Figure16a). The algorithm preforms the following steps to construct the NDP $\mathbb{P}(s,T)$:

1. Connect t_1 with the border node $b_1 = (b_{1_x}, t_{1_y}) \in Q_N$ horizontally using the path $P(b_1,t_1)$ as $(b_{1_x}, t_{1_y}) \to (b_{1_x}, t_{1_y} - 1) \to \cdots \to (t_{1_x}, t_{1_y})$.
2. Connect t_2 with the border node $b_2 = (b_{2_x}, t_{2_y}) \in Q_N$ horizontally using the path $P(b_2,t_2)$ as $(b_{2_x}, t_{2_y}) \to (b_{2_x}, t_{2_y} - 1) \to \cdots \to (t_{2_x}, t_{2_y})$. $P(b_1,t_1)$ and $P(b_2,t_2)$ are NDP because they are in parallel.
3. Apply Case 2 to connect the source node s with the border nodes $a_1 = b_1^E \in Q_S$ and $a_2 = b_2^E \in Q_S$ using the west and south paths. Figure16a shows all possibilities of nodes a_1 and a_2 (the dashed nodes). The length of $P(s,t_1)$ is at most equal to $W(t_1) + (2k-5)$ which occurs when $t_{1_y} = 3$. The length of $P(s,t_2)$ is at most equal to $W(t_2) + (2k-3)$ which occurs when $t_{2_y} = 2$. Also by applying Equation 3, the length of $P(s,t_1)$ is at lease equal to $W(t_1) + 3$ which occurs when $t_{1_y} = k-1$. The length of $P(s,t_2)$ is at least equal to $W(t_2) + 5$ which occurs when $t_{2_y} = k-2$.
4. Apply Case 2 to construct $P(s,t_3)$ and $P(s,t_4)$ using the north and east paths. The sum of lengths of $P(s,t_3)$ and $P(s,t_4)$ is $W(t_3) + W(t_4) + 1 \le |P(s,t_3)| + |P(s,t_3)| \le W(t_3) + W(t_4) + (2k-7)$. The upper and lower bounds occur when $t_{4_y} = 4$ and $t_{4_y} = k$ respectively.

It follows that $L(T) + 9 \le |\mathbb{P}(s,T)| \le L(T) + (6k - 15)$.

Case 10.2 Three destination nodes have $x = 0$:

Let $t_{1_x} = t_{3_x} = t_{4_x} = 0$, $t_{2_x} \ne 0$, and $t_{3_y} < t_{1_y} < t_{4_y}$ (see Figure16b). The NDP $\mathbb{P}(s,T)$ are:

1. $P(s,t_2)$ is same as the south path in Case 6. Figure16b shows all possibilities of the border node a in Q_S (the dashed nodes). The length of $P(s,t_2)$ is $W(t_2) + 1 \le P(s,t_2) \le W(t_2) + (2k-3)$. The upper bound occurs when $W(t_2) = 2$.
2. $P(s,t_1)$ is $(0,0) \to (-1,0) \to (-1,1) \to (-1,2) \to \cdots \to (-1, t_{1_y}) \to (t_{1_x}, t_{1_y})$. This path is valid because Q_W has no destination nodes and the y value of t_1 is at most equal to $k-1$. The length of $P(s,t_1)$ is equal to $W(t_1) + 2$.
3. $P(s,t_3)$ and $P(s,t_4)$ are same as the north and east paths in Case 2 respectively. The sum of lengths of $P(s,t_3)$ and $P(s,t_4)$ is $W(t_3) + W(t_4) + 1 \le |P(s,t_3)| + |P(s,t_3)| \le W(t_3) + W(t_4) + (2k-5)$. The upper and lower bounds occur when $t_{4_y} = 3$ and $t_{4_y} = k$ respectively.

It follows that $L(T) + 4 \le |\mathbb{P}(s,T)| \le L(T) + (4k - 8)$.

Case 10.3 Two destination nodes have $x = 0$:

Let $t_{3_x} = t_{4_x} = 0$ and $t_{1_x}, t_{2_x} \ne 0$ (see Figure16c). The algorithm performs the following steps to construct the NDP $\mathbb{P}(s,T)$:

1. Apply Case 2 to reach t_3 and t_4 using the north and east paths. The sum of lengths of $P(s,t_3)$ and $P(s,t_4)$ is $W(t_3) + W(t_4) + 1 \le |P(s,t_3)| + |P(s,t_3)| \le W(t_3) + W(t_4) + (2k-3)$. The upper and lower bounds occur when $t_{4_y} = 2$ and $t_{4_y} = k$ respectively.
2. Apply Case 7 to reach t_1 and t_2 using the west and south paths. The sum of lengths of $P(s,t_3)$ and $P(s,t_4)$ is $W(t_3) + W(t_4) + 2 \le$

TABLE 3: Specifying the 1^{st} and 2^{nd} Destination Nodes in Case 10.5

No. 1^{st} min-weight	No. 2^{nd} min-weight	Choose
1	1	1^{st} and 2^{nd} min-weight
1	>1	1^{st} min-weight and right of 2^{nd} min-weight
>1	≥ 0	1^{st} and 2^{nd} right of 1^{st} min-weight

TABLE 4: Specifying the 3^{rd} and 4^{th} Destination Nodes in Case 10.5

No. 1^{st} max-weight	No. 2^{nd} max-weight	Choose
1	1	1^{st} and 2^{nd} max-weight
1	>1	1^{st} max-weight and left of 2^{nd} max-weight
>1	≥ 0	1^{st} and 2^{nd} left of 1^{st} max-weight

$|P(s,t_3)|+|P(s,t_3)| \leq W(t_3)+W(t_4)+(4k-8)$. The upper bound occurs when one of these destination nodes is node $(1,1)$ and the weight value of the other is equal to three. The lower bound occurs when $W(t_3)=W(t_4)=k$.

It follows that $L(T)+3 \leq |\mathbb{P}(s,T)| \leq L(T)+(6k-11)$.

Case 10.4: One destination node has $x=0$:

Let $t_{4_x}=0$ and $t_{1_x}, t_{2_x}, t_{3_x} \neq 0$ (see Figure17a). Assume, without loss of generality, that:

1. t_3 is the min-weight/left destination node among t_1,t_2, and t_3.
2. t_1 and t_2 are respectively the top/left and bottom/right destination nodes only among t_1 and t_2.

The algorithm performs the following steps to construct the NDP $\mathbb{P}(s,T)$:

1. Apply Case 2 to construct $P(s,t_3)$ and $P(s,t_4)$ using the north and east paths. The sum of lengths of $P(s,t_3)$ and $P(s,t_4)$ is equal to $W(t_3)+W(t_4)$.
2. Apply Case 7 to construct $P(s,t_1)$ and $P(s,t_2)$ using the south and west paths. The sum of lengths of $P(s,t_1)$ and $P(s,t_2)$ is $W(t_1)+W(t_2)+2 \leq |P(s,t_1)|+|P(s,t_2)| \leq W(t_1)+W(t_2)+(4k-10)$. The upper and lower bounds occur when $W(t_1)=W(t_2)=3$ and $W(t_1)=W(t_2)=k$ respectively.

It follows that $L(T)+2 \leq |\mathbb{P}(s,T)| \leq L(T)+(4k-10)$.

Case 10.5 None of the destination nodes has $x=0$:

In this case, $t_{1_x}, t_{2_x}, t_{3_x}, t_{4_x} \neq 0$ (see Figure17b). The algorithm performs the following steps to construct the NDP $\mathbb{P}(s,T)$:

1. Count the number of destination nodes that their weight values are equal to the minimum weight among all destination nodes (1^{st} min-weight). For example in Figure17b, the number of destination nodes in the 1^{st} min-weight is one (t_4).
2. Count the number of destination nodes in the 2^{nd} min-weight among all destination nodes. For example in Figure17b, the number of destination nodes in the 2^{nd} min-weight is two (t_2 and t_3).
3. Use Table 3 to specify two destination nodes. Note that this Table specifies exactly two destination nodes. Let these destination nodes be t_3 and t_4. For example in Figure17b, Table 3 specifies the destination node in the 1^{st} min-weight (t_4) and the right destination node among those in the 2^{nd} min-weight (t_3).
4. Apply Case 2 to construct $P(s,t_3)$ and $P(s,t_4)$ using the north and east paths. The sum of lengths of $P(s,t_3)$ and $P(s,t_4)$ is equal to $W(t_3)+W(t_4)$.
5. Count the number of destination nodes that has the max-weight among all destination nodes (1^{st} max-weight). For example in Figure17b, the number of destination nodes in the 1^{st} max-weight is one (t_1).
6. Count the number of destination nodes in the 2^{nd} max-weight among all destination nodes. For example in Figure17b, the number of destination nodes in the 2^{nd} max-weight is two (t_2 and t_3).
7. Use Table 4 to specify two destination nodes. Note that these destination nodes are different than the destination nodes specified before. Let these destination nodes be t_1 and t_2. For example in Figure17b, Table 4 specifies the destination node in the 1^{st} max-weight (t_1) and the left destination node among those in the 2^{nd} max-weight (t_2).
8. Apply Case 7 to construct $P(s,t_1)$ and $P(s,t_2)$ using the south and west paths with wraparound links. The sum of lengths of $P(s,t_1)$ and $P(s,t_2)$ is $W(t_1)+W(t_2)+2 \leq$

TABLE 5: All subcases of Case 10

Case No.	Lower Bound	Upper Bound
10.1	$L(T)+9$	$L(T)+(6k-15)$
10.2	$L(T)+4$	$L(T)+(4k-8)$
10.3	$L(T)+3$	$L(T)+(6k-11)$
10.4	$L(T)+2$	$L(T)+(4k-10)$
10.5	$L(T)+2$	$L(T)+(4k-12)$

$|P(s,t_1)|+|P(s,t_2)| \leq W(t_1)+W(t_2)+(4k-12)$. The upper bound occurs when the weight value of one of these destination nodes is equal to three and the weight value of the other destination nodes is equal to four. The lower bound occurs when $W(t_3)=W(t_4)=k$.

It follows that $L(T)+2 \leq |\mathbb{P}(s,T)| \leq L(T)+(4k-12)$.

Table 5 shows the upper and lower bounds of all subcases of Case 10. Clearly, the minimum lower bound and the maximum upper bound occur when the cases are Case 10.4 (or 10.5) and Case 10.3 respectively. It follows that $L(T)+2 \leq |\mathbb{P}(s,T)| \leq L(T)+(6k-11)$. □

After showing how to construct the NDP for all 10 cases, the following theorem states that four NDP always exist in the Gaussian network G_k and provides the upper and lower bounds of the sum of the lengths of all four paths $|\mathbb{P}(s,T)|$.

Theorem 3.1. *In the Gaussian network G_k where k is the network diameter, let the source node be $s=(0,0)$ and the set of destination nodes be $T=\{t_j=(t_{j_x},t_{j_y})|1 \leq j \leq 4\}$. Then, there exist NDP $\mathbb{P}(s,T)$ such that the sum of the lengths of the paths in $\mathbb{P}(s,T)$ is*

$$L(T) \leq |\mathbb{P}(s,T)| \leq L(T)+(6k-11)$$

Proof. The Gaussian network G_k can be divided into four non-overlapped quadrants based on the source node's address. These quadrants are Q_N, Q_W, Q_S, and Q_E as defined in Section 3.1. The four destination nodes can be distributed in exactly $\binom{4+4-1}{4}=35$ ways represented as $\langle|Q_N|,|Q_W|,|Q_S|,|Q_E|\rangle$ where $|Q_i|$ is the number of destination nodes in quadrant i for $i=N,W,S,E$. To prove the theorem we need to show that the NDP exist for each one of these 35 cases. However, since G_k is vertex symmetric, constructing the NDP for only 10 cases is equivalent to constructing the NDP for the 35 cases. Table 1 shows the chosen 10 cases and the equivalent cases; the total is 35 cases. Now to prove the theorem we need to show that the NDP exist for each one of these 10 cases. Lemmas 3.1 to 3.10 prove that the NDP exist for the chosen 10 cases. Table 6 shows the upper and lower bounds of these 10 cases. It follows that $L(T) \leq |\mathbb{P}(s,T)| \leq L(T)+(6k-11)$. □

TABLE 6: All Cases

Case No.	Chosen Cases	Lower Bound	Upper Bound
1	$\langle 1,1,1,1\rangle$	$L(T)$	$L(T)$
2	$\langle 2,0,2,0\rangle$	$L(T)$	$L(T)+(4k-6)$
3	$\langle 2,2,0,0\rangle$	$L(T)+1$	$L(T)+(4k-6)$
4	$\langle 2,1,1,0\rangle$	$L(T)$	$L(T)+(2k-3)$
5	$\langle 2,0,1,1\rangle$	$L(T)+1$	$L(T)+(2k-2)$
6	$\langle 2,1,0,1\rangle$	$L(T)+1$	$L(T)+(2k-3)$
7	$\langle 3,0,0,1\rangle$	$L(T)+2$	$L(T)+(4k-6)$
8	$\langle 3,1,0,0\rangle$	$L(T)+1$	$L(T)+(4k-6)$
9	$\langle 3,0,1,0\rangle$	$L(T)+1$	$L(T)+(4k-5)$
10	$\langle 4,0,0,0\rangle$	$L(T)+2$	$L(T)+(6k-11)$

3.3. Time Complexity

The overall time complexity of the proposed algorithm equals the sum of time complexity of Step 1 and Step 2 (see Alg.1). In Step 1, the algorithm counts the number of destination nodes in each quadrants based on the addresses of the source and destination nodes. Clearly, this step can be done in a constant time $O(1)$.

In Step 2, the algorithm constructs the NDP by executing the procedure of one case out of 10 cases based on the number of destination nodes in each quadrants. Thus, the time complexity of Step 2 equals the time complexity of the most time consuming among the 10 cases.

To construct the NDP, the algorithm needs to know the left, right, top, bottom, max-weight, and min-weight destination nodes of a specific quadrants as defined in Definition 3.2. That requires sorting a number of destination nodes based on three criteria:

1. the x-coordinate to know the left and right destination nodes,
2. the y-coordinate to know the top and bottom destination nodes, and
3. the weight as defined in Equation 2 to know the max-weight, and min-weight destination nodes.

This sorting can be done using the bucket sort method. The time complexity of the bucket sort method equals the number of elements to be sorted multiplying by the number of sorting criteria. In the worst case,

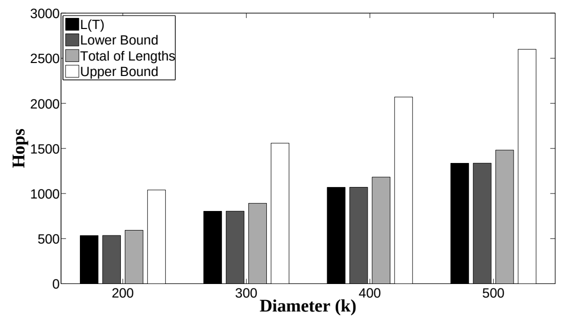


FIGURE 18: Shortest non-NDP vs. acutal NDP

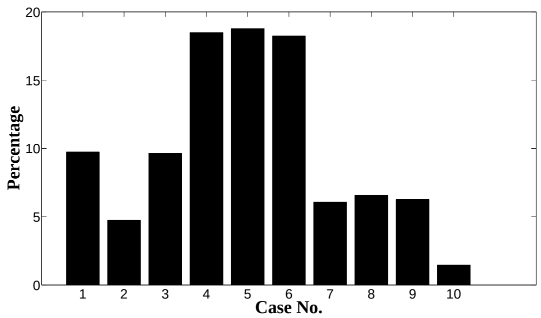


FIGURE 19: Distribution of Occurrence ($k = 500$, runs$= 10,000$)

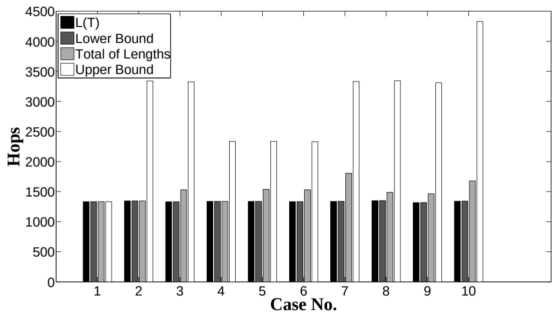


FIGURE 20: Case-wise shortest non-NDP vs. acutal NDP ($k = 500$, runs$= 10,000$)

the number of elements to be sorted equals four (the number of destination nodes) and that happens in Case 10. So, the time complexity of Step 2 equals $O(3*4) = O(12) \approx O(1)$. As a result, the overall time complexity of our proposed algorithm is a constant time $O(1)$.

4. SIMULATION RESULTS

In this section, we show the results of simulating the proposed algorithm. We mainly measure the sum of path lengths $|\mathbb{P}(s,T)|$ and compare it to the sum of destination nodes' weights $L(T)$ and the lower and upper bounds. The sum of destination nodes' weights $L(T)$ is equal to the sum of the shortest paths lengths where these paths are not necessarily node disjoint paths (NDP). Our simulation results show that all of the time the proposed algorithm gives NDP. The results also show that we need on the average about 10% more hops than the sum of destination nodes' weights $L(T)$ to construct the NDP in Gaussian networks.

We ran a simulator of the proposed algorithm 10,000 times for each one of the following networks: $G_{200}, G_{300}, G_{400}$, and G_{500}. In each run, the simulator *randomly* generated the four destination nodes T and the source node s. It returned the NDP $\mathbb{P}(s,T)$ for each run. After taking the averages, the results are shown in Figure18. In this figure, we compare the average number of hops of the sum of destination nodes' weights $L(T)$ and the sum of the actual NDP lengths $|\mathbb{P}(s,T)|$ along with the average of the lower and upper bounds. Clearly, the sum of the actual NDP lengths constructed by the proposed algorithm is very close to the sum of destination nodes' weights. In fact, the algorithm can construct the NDP with 10% more hops on the average than the sum of destination nodes' weights. This result is true regardless of the size of the network because the number of nodes in the network is irrelevant to the NDP construction process in the proposed algorithm. The 10% more hops is a small price to pay for earning the advantages of having the NDP.

For more clarification on why the difference between the actual NDP lengths and shortest distances is small, Figure19 shows the distribution of occurrence of each case for G_{500} over 10,000 runs. As shown in this figure, cases 4,5, and 6 are the most occurred cases with about 18% each. As shown in Table 6, the upper bounds of these cases are less than the other cases' upper bounds (except Case 1). Moreover, Case 10 which has the maximum upper bound occurs the least with 2% occurrence.

For more insights on the results, Figure20 compares for each case between the actual NDP lengths and shortest distances along with the lower and upper bounds for G_{500} over 10,000 runs. First, notice that the sum of the NDP lengths of cases 1, 2, and 4 is equal to the sum of the shortest path which is the lower bounds for these cases. That means the probability of the lower bound occurrence is almost 100% for these case which is expected especially for Case 1. Second, notice that the sum of the NDP lengths is far closer to the lower bound than the upper bound. That means the probability of the upper bound occurrence is very low for all cases.

5. CONCLUSION

Achieving high computing performance in parallel computing systems critically depends on constructing mutually node disjoint paths (NDP). In this work we provide and prove a novel algorithm to construct all possible NDP from a single source node to a set of destination nodes in the Gaussian interconnection networks. This algorithm construct four NDP which is equal to the maximum number of destination nodes.

We show that the sum of the NDP lengths constructed by the algorithm is bounded between the sum of the shortest paths and the this sum plus $(6k-11)$ where k is the diameter. We also show that the time complexity of the algorithm is constant $O(1)$. Finally, the simulation results show that on the average the sum of NDP lengths is 10% more than the sum of the shortest paths.

FUNDING

This work was supported by the National Science Foundation [CCF-1015804 to B.B., CNS-0846044 to B.H.].

ACKNOWLEDGEMENTS

The authors would like to thank King Saud University, Riyadh, Saudi Arabia for its support to Omar Alsaleh.

REFERENCES

[1] A. Bland, R. Kendall, D. Kothe, J. Rogers, and G. Shipman, "Jaguar: The world's most powerful computer," in *Proceedings of the Cray User Group Conference*, May 2009.

[2] N. Adiga and et al., "An overview of the bluegene/l supercomputer," in *Supercomputing, ACM/IEEE 2002 Conference*, nov. 2002, p. 60.

[3] K. Day and A. Tripathi, "A comparative study of topological properties of hypercubes and star graphs," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 5, no. 1, pp. 31 –38, Jan 1994.

[4] B. Parhami, *Introduction to parallel processing: algorithms and architectures.* Plenum Press, New York, 1999.

[5] J. Xu, *Topological Structure and Analysis of Interconnection Networks.* Kluwer Academic, Dordrecht, 2001.

[6] W. J. Dally and B. P. Towles, *Principles and practices of interconnection networks.* Morgan Kaufmann, 2004.

[7] J. Duato, S. Yalamanchili, and L. Ni, *Interconnection networks.* Morgan Kaufmann, 2003.

[8] L. Bhuyan and D. Agrawal, "Generalized hypercube and hyperbus structures for a computer network," *Computers, IEEE Transactions on*, vol. C-33, no. 4, pp. 323 –333, Apr 1984.

[9] P. Corbett, "Rotator graphs: an efficient topology for point-to-point multiprocessor networks," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 3, no. 5, pp. 622 –626, Sep 1992.

[10] Y. Li and S. Peng, "Dual-cubes: a new interconnection network for high- performance computer clusters," in *Proceedings of the 2000 International Computer Architecture*, 2000, pp. 51–57.

[11] N. Adhikari and C. Tripathy, "Folded dual-cube: A new interconnection topology for parallel systems," in *International Conference on Information Technology, ICIT '08*, Dec 2008, pp. 75 –78.

[12] R. Wu, G. Chen, Y. Kuo, and G. Chang, "Node-disjoint paths in hierarchical hypercube networks," *Information Sciences*, vol. 177, no. 19, pp. 4200 – 4207, 2007.

[13] A. Bossard, K. Kaneko, and S. Peng, "Node-to-set disjoint-path routing in perfect hierarchical hypercubes," in *Proceedings of the International Conference on Computational*, vol. 4, no. 0, 2011, pp. 442 – 451.

[14] A. Bossard and K. Kaneko, "Set-to-set disjoint-path routing in perfect hierarchical hypercubes," in *Proceedings of The Fourth International Conference on Computer Science and Software Engineering*, 2011, pp. 51–57.

[15] K. Kaneko and S. Peng, "Node-to-set disjoint paths routing in dual-cube," in *Parallel Architectures, Algorithms, and Networks*, May 2008, pp. 77 –82.

[16] A. Bossard, K. Kaneko, and S. Peng, "Node-to-set disjoint-path routing in metacube," in *International Conference on Parallel and Distributed Computing, Applications and Technologies*, Dec 2009, pp. 57 –62.

[17] C.-N. Lai, G.-H. Chen, and D.-R. Duh, "Constructing one-to-many disjoint paths in folded hypercubes," *Computers, IEEE Transactions on*, vol. 51, no. 1, pp. 33–45, Jan 2002.

[18] D. Liu and J. Li, "Many-to-many n-disjoint path covers in n-dimensional hypercubes," *Inf. Process. Lett.*, vol. 110, no. 0, pp. 580–584, July 2010.

[19] Y.-K. Shih and S.-S. Kao, "One-to-one disjoint path covers on k-ary n-cubes," *Theoretical Computer Science*, vol. 412, no. 35, pp. 4513–4530, 2011.

[20] O. Sinanoglu, M. Karaata, and B. AlBdaiwi, "An inherently stabilizing algorithm for node-to-node routing over all shortest node-disjoint paths in hypercube networks," *Computers, IEEE Transactions on*, vol. 59, no. 7, pp. 995–999, 2010.

[21] Y. Xiang and I. A. Stewart, "One-to-many node-disjoint paths in (n, k)-star graphs," *Discrete Applied Mathematics*, vol. 158, no. 1, pp. 62–70, 2010.

[22] X.-B. Chen, "Paired many-to-many disjoint path covers of the hypercubes," *Information Sciences*, vol. 236, no. 0, pp. 218 – 223, 2013.

[23] S. Ling and W. Chen, "Node-to-set disjoint paths in biswapped networks," *The Computer Journal*, 2013.

[24] C.-N. Lai, "Optimal construction of all shortest node-disjoint paths in hypercubes with applications," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 23, no. 6, pp. 1129–1134, 2012.

[25] L. LiptáK, E. Cheng, J.-S. Kim, and S. W. Kim, "One-to-many node-disjoint paths of hyper-star networks," *Discrete Applied Mathematics*, 2012.

[26] K. Kaneko and Y. Suzuki, "An algorithm for node-to-set disjoint paths problem in rotator graphs," *IEICE TRANSACTIONS on Information and Systems*, vol. 84, no. 9, pp. 1155–1163, 2001.

[27] K. KANEKO and Y. SUZUKI, "Node-to-set disjoint paths problem in pancake graphs," *IEICE TRANSACTIONS on Information and Systems*, vol. 86, no. 9, pp. 1628–1633, 2003.

[28] S. Peng and K. Kaneko, "Set-to-set disjoint-path routing in metacube," in *Parallel and Distributed Computing, Applications and Technologies (PDCAT), 2010 International Conference on*, 2010, pp. 128–137.

[29] K. Kaneko and S. Peng, "Set-to-set disjoint paths routing in dual-cubes," in *Parallel and Distributed Computing, Applications and Technologies, 2008. PDCAT 2008. Ninth International Conference on*, 2008, pp. 129–136.

[30] Q.-P. Gu and S. Peng, "Fault tolerant routing in toroidal networks," in *Parallel Algorithms/Architecture Synthesis, 1995. Proceedings. First Aizu International Symposium on*, 1995, pp. 162–168.

[31] M. Flahive and B. Bose, "The topology of gaussian and eisenstein-jacobi interconnection networks," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 21, no. 8, pp. 1132 –1142, aug. 2010.

[32] C. Martinez, R. Beivide, E. Stafford, M. Moreto, and E. Gabidulin, "Modeling toroidal networks with the gaussian integers," *Computers, IEEE Transactions on*, vol. 57, no. 8, pp. 1046 –1056, aug. 2008.

[33] Z. Zhang, Z. Guo, and Y. Yang, "Efficient all-to-all broadcast in gaussian on-chip-networks," *Computers, IEEE Transactions on*, vol. PP, no. 99, p. 1, 2012.

[34] C. Martnez, E. Vallejo, R. Beivide, C. Izu, and M. Moret, "Dense gaussian networks: Suitable topologies for on-chip multiprocessors," *International Journal of Parallel Programming*, pp. 193–211, 2006.