A THEORETICAL NETWORK MODEL AND THE INCREMENTAL
HYPERCUBE-BASED NETWORKS

DISSERTATION

Presented to the Graduate Council of the
University of North Texas in Partial
Fulfillment of the Requirements

For the Degree of

DOCTOR OF PHILOSOPHY

By

Ai-sheng Mao, B.S., M.S.
Denton, Texas
May, 1995
A THEORETICAL NETWORK MODEL AND THE INCREMENTAL HYPERCUBE-BASED NETWORKS

DISSERTATION

Presented to the Graduate Council of the University of North Texas in Partial Fulfillment of the Requirements

For the Degree of

DOCTOR OF PHILOSOPHY

By

Ai-sheng Mao, B.S., M.S.
Denton, Texas
May, 1995
Mao, Ai-sheng, A Theoretical Network Model and the Incremental Hypercube-Based Networks. Doctor of Philosophy (Computer Science), May, 1995, 164 pp., 8 tables, 52 illustrations, bibliography, 81 titles.

The study of multicomputer interconnection networks is an important area of research in parallel processing. We introduce vertex-symmetric Hamming-group graphs as a model to design a wide variety of network topologies including the hypercube network. A Hamming-group graph $HGG = (V, \Omega)$ is a transformation graph for which node-set $V = \{i \mid 0 \leq i < 2^n \text{ and } n \geq 1\}$ is the Hamming group which contains all binary strings of length $n$. A generator, a bit string of length $n$, acting on the Hamming group is defined as the bitwise Exclusive-OR operation. The concept of incremental Hamming-group graphs enhances the network design with unit incremental capability. In particular, this enhanced model provides an unifying framework for generating many possible supergraphs of incomplete hypercubes which can have an arbitrary number of nodes. From our models, we derive and analyze two new families of succinctly representable and labeled networks, called the Hamming cubes and the enhanced generalized incomplete hypercubes.

The Hamming cube ($HC$) networks can recursively grow from the existing ones with the increment of one node at a time, have half of logarithmic diameter, and are easily decomposable. A simple oblivious (or non-adaptive) routing scheme is designed, by which the routing paths have the optimal length bounded by the network diameter. Hamming cubes are shown to be optimally fault-tolerant, strongly resilient, and they exhibit very good performance even under multiple faults. The reliability and fault-tolerance of $HC_n$, the $n$-dimensional Hamming cube of $2^n$ nodes, are better than those of the binary hypercube ($Q_n$). We design a testing algorithm for a faulty $HC_n$ and
our diagnostic algorithm can identify up to \( n + 1 \) faulty processors. Taking advantages of enhanced edges and recursive nature, the average distance, average-distance-degree cost, and message traffic density of \( HC_n \) are all less than those parameters of \( Q_n \). Furthermore, \( HC_n \) has constant vulnerability and can be laid out in an \( O(N \times N) \) square, where \( N = 2^n \).

An enhanced generalized incomplete hypercube (EGIQ) can be viewed as an “enhanced” incomplete hypercube with extra links or a “generalized” folded hypercube with incrementability of one. This proposed family of networks also has half of logarithmic diameter. Simple deterministic routing scheme is designed and the lengths of routing paths are shown to be bounded by the diameter. Due to additional enhanced links, both the networks \( EGIQ(N) \) and \( HC(N) \) of an arbitrary order \( N \) have improved values of network parameters compared to the incomplete hypercubes of the same size.

In addition to the routing schemes (i.e., one-to-one communication), we design the broadcasting (i.e., one-to-all communication) schemes for the proposed networks by constructing two types of embedded directed broadcasting trees — spanning trees and multiple spanning trees (MUST’s) — from arbitrary source nodes. A MUST is composed of at least one edge-disjoint spanning tree. An analysis of the time complexities of broadcasting schemes under the one-port and all-port communication models concludes that Hamming cubes and enhanced generalized incomplete hypercubes provide more efficient communication network topologies than the complete or incomplete binary hypercubes.

The embeddability of Hamming cubes is also explored in this dissertation. It is shown that several important structures including Hamiltonian path and cycle, complete binary tree and its variants, and tree machine can be optimally embedded
into Hamming cubes in a one-to-one fashion and with minimum expansion. For example, Hamming cubes are pancyclic networks, that is the cycles of all lengths can be embedded. A complete binary tree is a subgraph of the Hamming cube with the same size, which is a distinct advantage over binary hypercubes. Also, X-tree, hypertree, full-ringed, and half-ringed binary trees are subgraphs of Hamming cubes with unit expansion cost. Tree machines can be embedded into Hamming cubes and the generalized incomplete hypercubes with expansion approximately equal to one, while both dilation and edge congestion are equal to two. However, dilation-1 embedding of tree machines into Hamming cubes require an expansion of $\frac{7}{6}$, another advantage over the incomplete hypercubes which requires a dilation of two.
ACKNOWLEDGEMENTS

I would like to express my deepest appreciation to my advisor, Dr. Sajal K. Das, for his counsel, support, and friendship given to me. I would also like to thank the members of my committee, Dr. Neal Brand, Dr. Roy T. Jacob, and Dr. Farhad Shahrokhi for their time and suggestions. This research is supported by Texas Advanced Research and Technology Program Grants TARP-003594003 and TATP-003594031, and by Department of Computer Science at the University of North Texas. I also thank the faculty of the department for their teaching.

I would like to dedicate this dissertation to my wife and my family for their love and support. Without them, this dissertation would not have been possible.
# TABLE OF CONTENTS

**LIST OF TABLES**

**LIST OF ILLUSTRATIONS**

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1</td>
<td>Hypercube-based Networks</td>
<td>3</td>
</tr>
<tr>
<td>1.2</td>
<td>Efficient Communication</td>
<td>4</td>
</tr>
<tr>
<td>1.3</td>
<td>Network Embeddings</td>
<td>5</td>
</tr>
<tr>
<td>1.4</td>
<td>Research Motivations</td>
<td>5</td>
</tr>
<tr>
<td>1.5</td>
<td>Organization of Chapters</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>PRELIMINARIES ON NETWORKS</td>
<td>11</td>
</tr>
<tr>
<td>2.1</td>
<td>Definition and Notations</td>
<td>11</td>
</tr>
<tr>
<td>2.2</td>
<td>Communication Models</td>
<td>12</td>
</tr>
<tr>
<td>2.3</td>
<td>Embedding Costs</td>
<td>13</td>
</tr>
<tr>
<td>2.4</td>
<td>Embedding Guest Network Structures</td>
<td>14</td>
</tr>
<tr>
<td>2.5</td>
<td>Hypercube-family of Networks</td>
<td>16</td>
</tr>
<tr>
<td>2.6</td>
<td>Enhanced Hypercube-family of Networks</td>
<td>17</td>
</tr>
<tr>
<td>2.7</td>
<td>Hypercube-like family of Networks</td>
<td>18</td>
</tr>
<tr>
<td>3</td>
<td>THEORETICAL NETWORK MODEL</td>
<td>22</td>
</tr>
<tr>
<td>3.1</td>
<td>Hamming Group</td>
<td>22</td>
</tr>
<tr>
<td>3.2</td>
<td>Hamming Group Graphs</td>
<td>23</td>
</tr>
<tr>
<td>3.3</td>
<td>Properties of Hamming Group Graphs</td>
<td>26</td>
</tr>
<tr>
<td>3.4</td>
<td>Incremental Hamming Graphs</td>
<td>27</td>
</tr>
<tr>
<td>3.5</td>
<td>Types of Hamming Generators</td>
<td>28</td>
</tr>
<tr>
<td>3.6</td>
<td>Summary</td>
<td>30</td>
</tr>
<tr>
<td>4</td>
<td>THE HAMMING CUBE NETWORKS</td>
<td>31</td>
</tr>
<tr>
<td>4.1</td>
<td>Network Definition</td>
<td>31</td>
</tr>
<tr>
<td>4.2</td>
<td>Topological Properties</td>
<td>34</td>
</tr>
<tr>
<td>4.3</td>
<td>Network Comparison</td>
<td>39</td>
</tr>
<tr>
<td>4.4</td>
<td>Decomposition of Hamming cubes</td>
<td>41</td>
</tr>
<tr>
<td>4.5</td>
<td>Simple Routing Schemes</td>
<td>44</td>
</tr>
<tr>
<td>4.6</td>
<td>Fault-tolerance and Reliability Analyses</td>
<td>46</td>
</tr>
<tr>
<td>4.6.1</td>
<td>Optimal Fault-tolerance</td>
<td>46</td>
</tr>
<tr>
<td>4.6.2</td>
<td>Node-Disjoint Paths</td>
<td>52</td>
</tr>
<tr>
<td>4.6.3</td>
<td>Two-terminal Reliability</td>
<td>56</td>
</tr>
<tr>
<td>4.6.4</td>
<td>Container Quality</td>
<td>58</td>
</tr>
</tbody>
</table>
10.1.2 Hamiltonian Cycles in $HC_n$ .................................................. 136
10.1.3 Hamiltonian Cycles in $HC(N)$, where $2^{k-1} < N < 2^k$ and $k \geq 3$. 137
10.2 Embedding Complete Binary Trees and Their Related .................. 143
10.3 Embedding Tree Machine .............................................................. 147
10.4 Summary ....................................................................................... 151

11 CONCLUSIONS ..................................................................................... 153

10.1 Future Work ................................................................................... 157

BIBLIOGRAPHY ..................................................................................... 159
LIST OF TABLES

4.1 Comparison of several hypercube-like networks. ...................................... 40
4.2 Comparison of $IQ^k_n$ and $HC(2^n + 2^k)$ ........................................ 41
4.3 Comparison of $IQ^k_{n-1}$ and $HC(2^n + 2^{n-1})$ ................................. 41
4.4 The number of disjoint paths for Example 4.8. .................................... 56
7.1 Time complexity of communication ......................................................... 111
9.1 $m^2_{i,j}$, where $0 < i < 5$ and $2 < j < 10$. ........................................ 125
9.2 $m^4_{i,j}$, where $0 < i < 5$ and $4 < j < 10$. ........................................ 126
10.1 Embedding comparisons of hypercube-based networks. .......................... 158
LIST OF ILLUSTRATIONS

1.1 Classification of Parallel Computers ........................................... 2
2.1 (a) Two-rooted complete binary tree TCBT(3). (b) Full-ringed tree FRT(3). ......................................................... 14
2.2 (a) Hypertree HT(4). (b) The 4-dimensional tree machine TM(4). .......................... 15
2.3 (a) The incomplete hypercube IQ4. (b) The 4-dimensional binary hypercube Q4. ............. 16
2.4 (a) The enhanced incomplete hypercube EIQ4. (b) The 4-dimensional folded hypercube FQ4. .............................................. 18
2.5 (a) A twisted 4-cube TQ4. (b) The 4-dimensional cossed cube CC4. ................................. 19
2.6 (a) The incrementally extensible hypercube IEQ(11) of order 11. (b) The supercube SC(11) of order 11. ............................................. 20
3.1 Hamming group graphs HGG = (V, Ω), where set V = {i | 0 ≤ i < 2n} and (a) Ω = Ω1, (b) Ω = Ω1 ∪ Ω2, (c) Ω = Ω1 ∪ Ω3, and (d) Ω = Ω1 ∪ Ω2 ∪ Ω3. .................. 25
3.2 Incremental Hamming graphs IHG(N) with (a) Ω = Ω1 and (b) Ω = Ω1 ∪ Ω2, where 2 ≤ N ≤ 8. ............................................ 27
3.3 Interconnection network model. .................................................. 29
4.1 Hamming cubes, HC(N) for 2 ≤ N ≤ 8 and N = 16. ........................................... 32
4.2 A binomial spanning tree of HC(16), rooted at node v0, using the E2-edges. .................. 33
4.3 (a) Decomposition of HC3 = HC(8) with relabeling of nodes. (b) Decomposition of HC(12) with node relabeling. (c) A decomposition of HC(15). .................................................. 42
4.4 In network HC5, the node-disjoint paths between (a) v1 and v3, (b) v2 and v4, and (c) v0 and v7. ............................................ 55
4.5 Two-terminal reliabilities of HCn and Qn with p = 0.8. ........................................... 58
4.6 Two-terminal reliabilities of HCn and Qn with p = 0.6. ........................................... 59
4.7 Examples of the diagnostic algorithm. .......................................... 64
5.1 Enhanced generalized incomplete hypercubes, EGIQ(N) for 2 ≤ N ≤ 8 and N = 11. ........... 69
5.2 Network decompositions of EGIQ(11) and EGIQ(15). .......................................... 74
6.1 The BST's rooted at nodes v0 and v12 respectively in Q4. ......................................... 80
6.2 The STGIQ(N)(15) rooted at node v11. ............................................. 82
6.3 The MUST of Q4. ................................................................. 84
6.4 The (virtual) source nodes in GIQ(11) and GIQ(27). ............................................. 86
6.5 The MUST of Q4 rooted at node v4. ........................................................................ 92
6.6 The MUSTGIQ(27) rooted at node v22. ..................................................................... 93
7.1 The B+ST's in HC4 and HC5. ........................................................................ 97
7.2 The STHC4 rooted at node v10. ........................................................................ 101
7.3 (a) The STHC(42) rooted at node v19. (b) The B+ST rooted at node v3. 104
7.4 The MUST of FQ4 and HC4. ........................................................................ 107
7.5 The MUSTHC(27) rooted at node v22. ..................................................................... 110
8.1 The SBT rooted at node s in FQ5. ........................................................................ 114
8.2 Example of $ST_{EGIQ(18)}$ for the case $s \in X_p$. 117
8.3 The MUST rooted at $v_9$ in $EGIQ(11)$. 120
9.1 The average distances of $d_{HC_n}$, $d_{Q_n}$, and $d_{FQ_n}$. 127
9.2 The average distances $d_{EGIQ(N)}$, $d_{HC(N)}$, and $d_{GIQ(N)}$. 128
9.3 The cost of networks $HC_n$, $Q_n$, and $FQ_n$. 129
9.4 The average-distance-degree costs of networks $EGIQ(N)$, $HC(N)$, and $GIQ(N)$. 130
9.5 The MTD's of networks $HC_n$, $Q_n$, and $FQ_n$. 131
9.6 The MTD’s of networks $EGIQ(N)$, $HC(N)$, and $GIQ(N)$. 132
10.1 An embedding of Hamiltonian cycle $C(v_9, S_{20n,1})$ in $HC_4$. 136
10.2 An embedding of a Hamiltonian cycle in $HC(11)$. 139
10.3 The embedding of Hamiltonian cycles in $HC_3$, $GIQ(6)$, and $HC(14)$. 141
10.4 An embedding of Hamiltonian cycles in $HC(15)$. 142
10.5 The embedding of complete binary tree of height 4 in $HC_5$. 144
10.6 The embedding of tree-cube $TC(3)$ rooted at node $v_9$ in $HC_4$. 146
10.7 The embedding of hypertree $HT(4)$ of height 4 in $HC_5$. 147
10.8 The embedding of full-ring tree of height 3 in $HC_4$. 148
10.9 The embedding of half-ring tree of height 3 in $HC_4$. 149
10.10 The embedding of $TM(3)$ in $HC(28)$. 150
10.11 The upper tree of $TM(3)$ in $HC(28)$. 151
10.12 The embedding of $TM(4)$ in $HC(56)$. 152
CHAPTER 1

INTRODUCTION

The increasing demand for high-speed reliable computing motivates the study of massively parallel machines. The multiple-instruction stream multiple-data stream (MIMD) computers can be classified into two categories: shared memory machines (multiprocessors) and non-shared or distributed memory machines (multicomputers), as shown in Figure 1.1. In loosely coupled multiprocessors and distributed memory multicomputers, each processing unit is connected to others by an interconnection network and process cooperation occurs through message passing.

A static interconnection network for multicomputer systems can be modeled as an undirected graph $G = (V, E)$, where $V$ is the node-set representing processor-memory modules and $E$ is the edge-set representing the communication links among processors. Several well-known graphs have been studied as the underlying topologies of the interconnection networks, for example, linear arrays, rings, meshes, trees [9, 33, 43, 44, 53], binary hypercubes (also called n-cube) [7, 13], cube-connected cycles [58], pyramids [68], shuffle exchange [67], and so on. For general surveys of interconnection networks, refer to [38, 54, 60].

Several other topologies have been proposed recently for interconnection networks, all of which can be classified as follows by their design philosophies:

1. Mathematical networks: These are designed from a particular theory that defines the adjacency of nodes. The involved mathematical theory and the corresponding networks include, to name a few,

   - Group theory: Group graphs [3, 5], Star and Pancake networks [1, 2, 59].
MIMD Computer

Multiprocessor (Shared Memory)

Tightly Coupled
- Shared Bus:
  - Encore’s Multimax
  - Sequent’s Symmetry S/81

Loosely Coupled
- Carnegie-Mellon’s Cm*

Multicomputer (Distributed Memory)

- Ametek’s S/14
- Intel’s iPSC/860
- NCUBE’s NCUBE/10
- Marpar MP–2
- Connection Machine CM–2

Crossbar Switch:
- Carnegie-Mellon’s C.mmp

Multistage Switching Network:
- BBN Butterfly GP/1000
- New York University’s Ultracomputer

Figure 1.1: Classification of Parallel Computers

- **Combinatorics:** Pascal graphs [32], de Bruijn networks [10, 62], Stirling networks [24], and recursive combinatorial networks [27, 28].

- **Others:** Fibonacci cubes [45].

2. **Enhanced networks:** These are improved from existing networks by augmenting their abilities with various design and performance considerations. For example, twisted n-cube [37], hyernet [46], enhanced hypercubes [76], folded hypercubes [36], incomplete hypercubes [48, 74], enhanced incomplete hypercubes [19], incrementally extensible hypercubes [70], hyperbanyan [39], crossed cube [35], supercube [8, 64], generalized supercube [65], bridged hypercubes [22], and generalized incomplete hypercubes [26, 72] belong to this category.
3. **Hybrid networks**: They combine two or more families of networks through some graph operations (e.g. direct product) or the definitions of node adjacency. Hyper Peterson [23], Folded Peterson [56], hyper-de Bruijn [40], banyan-hypercube [81], de Bruijn cube [17], extended hypercubes [51], and hierarchical cubes [41] networks fall in this category.

Some of the desirable characteristics of multicomputer interconnection networks are small degree, small diameter, simple routing scheme, efficient broadcasting schemes, high degree of symmetry, incremental expansion (ideally unit incrementability), high fault-tolerance (or connectivity), and embeddability of other networks. These properties are related to preformance parameters such as communication delays or path-lengths, reliability, congestion, throughput, or special design considerations such as expandability, regularity, and reconfiguration. Since many of these factors make contradictory demands, a compromise is necessary. Therefore, search for better interconnection networks with specified properties still continues.

1.1. **Hypercube-based Networks**

Among the existing networks, binary hypercubes have received significant attention because of their attractive characteristics: node- and edge-symmetries, logarithmic diameter and average distance, high fault-tolerance, scalability, simple routing and broadcasting schemes, and efficient mapping of various guest networks and application problems. Several commercial and research systems are based on this network architecture, for example, the Cosmic cube [63], the Intel iPSC systems, the Connection Machine (CM-1/2) [73] and among others.

However, the hypercube topology has the drawback that its size grows as a power of two, implying that the incremental capability is poor. Therefore, two variants,
namely incomplete hypercubes [48] consisting of $2^n + 2^k$ nodes, where $0 \leq k < n$, and generalized incomplete hypercubes [26, 72] having $N \geq 1$ nodes, were introduced with incrementabilities of $2^k$ and 1, respectively. These families of networks belong to the so-called hypercube-family. It is easy to see that the generalized incomplete hypercubes have poor fault-tolerance.

Recent efforts have been made to improve the performance of (incomplete) hypercubes with additional links, leading respectively to folded hypercubes [36] and enhanced incomplete hypercubes [19]. These two networks (among others) can be categorized into the enhanced hypercube-family.

In our classification of the recently proposed networks (Section 1.1), there are several enhanced networks commonly considering the hypercube topology as the fundamental structure to design new networks according to the design options. For example, twisted $n$-cube [37] and crossed cube [35] recursively twist a pair of edges in a shortest cycle (consisting of four nodes) of the binary hypercubes. Incrementally extensible hypercubes [70] use a different node-labeling mechanism to define their enhanced edges among several constitutive subcubes. A supercube [8, 64] divides its node-set into three subsets and add the enhanced edges to a pair of nodes in the subsets whose Hamming distance is two. These networks will be referred to as the hypercube-like family in our discussion.

1.2. Efficient Communication

Efficient communication plays an important role in the performance of any multicompouer system. The routing (one-to-one communication) schemes for the (enhanced) hypercube-family of networks have been proposed in [36, 47, 48, 69, 78].

Broadcasting (one-to-all communication) is one of the most important commu-
communication primitives in a network. Two types of embedded broadcasting topologies — the spanning trees (ST’s) and multiple spanning trees (MUST’s) — have been constructed in the hypercube-family and folded hypercubes [36, 47, 71, 72]. Both the trees are directed and have the property of edge-disjointness in the sense of directed edges, which can be realized by the assumption that a communication link is full-duplex. The broadcasting data is sent from the root (source node) to the other nodes (destination nodes) of the trees following the direction of edges. Under the one-port and all-port communication models [47], the broadcasting due to these two types of embedded trees can speed up the time complexity.

1.3. Network Embeddings

The mapping of one network architecture into another provides a geometrical abstraction between two different network topologies. Through such network embeddings, the parallel algorithms developed for the original architecture can be directly transferred with little efforts to another new one. The hypercube network has been proven that it has attractive embeddability [54] and hence, can efficiently simulate many other networks within a small factor of slowdown. For the hypercube-family of networks, there are several networks that can be optimally embedded such as linear array and rings [15, 18, 61, 66], trees and their related variants [11, 18, 34, 57, 75, 80], and meshes or grids [14, 20, 61].

1.4. Research Motivations

From the preceding literature review of the hypercube-based networks, the following questions can be arisen:
• Is there any relation underlying the design of additional edges in the enhanced hypercube-family networks? What is the general interpretation of the network topologies for the hypercube-like family?

To answer the preceding questions, we will consequently resort to a network model which not only can classify the existing hypercube-based networks in an unified way, but also can generate many new topologies which are of interest to the designers of multicomputer interconnection networks.

The fact is that the generalized incomplete hypercubes have the merits of unit incrementability, but have the poor fault-tolerance. To improve this weakness, the approach of adding enhanced links can be applied. Closely comparing the members in the enhanced hypercube-family with those in the hypercube-family, it can be concluded that the enhanced version of the generalized incomplete hypercubes has not yet been designed. From this special perspective, we have the following questions:

• Can we design a hypercube-based family of networks which satisfy many desirable properties especially including incremental expansion, high fault-tolerance, small diameter, and so on?

In addition to the structural properties, more crucial questions can be further asked for a topology of interconnection networks:

• What is the performance and reliability of the network? Does the network topology provide a mechanism for efficient communication? What is the embeddability of the network?

In this dissertation, we will answer to these questions by designing communication efficient networks which support unit incrementability as well as provide low-cost
embedding of various computation structures.

1.5. Organization of Chapters

The remainder of this dissertation is organized into ten chapters. Chapter 2 gives the definitions and notations used thoroughly in this dissertation. Chapter 3 introduces a network model, called the Hamming group graphs [30]. The Hamming group graphs are the transformation graphs which use the Hamming-group as generators, and can be used to generate many important classes of hypercube-like topologies. This model enhanced with the unit incremental capability provides a framework for generating many possible supergraphs of incomplete hypercubes having an arbitrary number of nodes.

In particular, we derive from our model two new families of succinctly representable and labeled networks, called the Hamming cubes (HC's) and the enhanced generalized incomplete hypercubes (EGIQ's) [25]. Both families belong to the class of incremental Hamming graphs.

Chapter 4 provides the definition of the Hamming cubes and explores their structural properties. The Hamming cube networks can recursively grow from the existing ones with the increment of one node at a time. This family of networks has half of logarithmic diameter, and is easily decomposable. Simple oblivious (or non-adaptive) routing schemes in which the routing path of a message depends only upon its source and destination nodes are designed for Hamming cubes. The routing paths have the optimal length bounded by the diameter of HC's. Hamming cubes are optimally fault-tolerant since the node-connectivity is equal to the minimum degree. Especially, the connectivity is $n + 1$ for the $n$-dimensional Hamming cubes $HC_n$ consisting of $2^n$ nodes.
We also evaluate the performance of Hamming cubes by deriving several important parameters [29]. For n-dimensional Hamming cubes, average distance, two-terminal reliability (also called path reliability), fault-node diameter, container, and vulnerability are explored and compared to those of binary hypercubes and folded hypercubes. The VLSI layout and diagnostic algorithm for detecting the faulty nodes in the injured n-dimensional Hamming cubes are also considered.

Chapter 5 presents another family of networks, the enhanced generalized incomplete hypercubes. This family can be viewed as an “enhanced” generalized incomplete hypercube [26, 72] with extra links or a “generalized” folded hypercube [36] with incrementability of one. The enhanced generalized incomplete hypercubes also have half of logarithmic diameter and can be easily decomposable. Simple deterministic schemes whose routing paths have the length of at most the diameter are designed for the networks.

In Chapter 6, we design the broadcasting (one-to-all communication) schemes for generalized incomplete hypercubes by constructing two types of embedded broadcasting trees, the spanning trees (STGIQ’s) and multiple spanning trees (MUSTGIQ’s) [26]. The broadcasting according to these embedded trees is evaluated for one-port and all-port communication models. The results explored in this chapter will be used for the discussion in Chapters 7 and 8 which construct broadcasting trees in Hamming cubes and enhanced generalized incomplete hypercubes.

Chapter 7 discusses the broadcasting schemes for Hamming cubes by constructing the embedded broadcasting trees [31] and evaluate their time complexity by using the one-port and all-port communication models.

The tree construction depends on the size of Hamming cubes HC’s and the identification of the source node. The spanning tree $ST_{HC}$ in the n-dimensional
Hamming cube \((HC_n)\) is built from a newly defined building block, called the binomial plus spanning tree \((B^+ST)\), which is a variant of the well-known binomial spanning tree \((BST)\) [47]. For a Hamming cube \(HC(N)\) of arbitrary nodes, the spanning tree \(ST_{HC(N)}\) is constructed with the help of \(ST_{HC_n}, BST\) and \(B^+ST\).

Since an \(n\)-dimensional folded hypercube \((FQ_n)\) is a spanning subgraph of the \(n\)-dimensional Hamming cube \(HC_n\) [30], the multiple spanning tree in \(FQ_n\) also provide a multiple spanning tree in \(HC_n\). Thus, we will concentrate on the construction of multiple spanning tree \(MUST_{HC(N)}\) in the Hamming cube \(HC(N)\) of an arbitrary size \(N\). The technique for the construction of multiple spanning trees in generalized incomplete hypercubes presented in Chapter 6 can be used for this purpose.

In Chapter 8, we further design the broadcasting schemes for enhanced generalized incomplete hypercubes by using the embedded broadcasting trees. The spanning broadcasting trees \((SBT's)\) in folded hypercubes and the concept of link-replacement help the construction of our spanning trees. For the multiple spanning trees, the algorithm which constructs the \(MUST_{HC(N)}\) in Hamming cubes can be used with little modification.

Chapter 9 evaluates the performance of Hamming cubes and enhanced generalized incomplete hypercubes of arbitrary size (not a power of two). Three parameters — average distance, network cost, and message traffic density — are empirically analyzed and compared to those of generalized incomplete hypercubes.

Chapter 10 shows the embeddability of Hamming cubes. Several guest networks such as Hamiltonian cycle, ring, complete binary tree, and hypertree are shown to be optimally embedded in Hamming cubes with dilation one and minimum expansion. Most importantly, due to the recursive nature and additional enhanced edges of Hamming cubes, the embeddings are improved and better than those for incom-
plete hypercubes and binary hypercubes. For example, it is shown that the complete binary tree of height \( n \) cannot be embedded in the \( n \)-dimensional binary hypercube, but can be embedded in the \( n \)-dimensional Hamming cube.

Chapter 11 concludes the dissertation with a summary of the presented works and suggests future research in this area.
CHAPTER 2

PRELIMINARIES ON NETWORKS

In this chapter, we provide the definition of networks and notations that will be used in our study.

2.1. Definition and Notations

An interconnection network can be modeled as an undirected graph $G = (V, E)$, for which $V(G)$ is the node-set representing processing elements and $E(G)$ is the edge-set representing the communication links among them. Thus, when we use the word "network" in our context, it means the underlying graph topology of the network. A graph $H$ is a subgraph of $G$ if $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. Whenever $V(H) = V(G)$, then $H$ is called a spanning subgraph of $G$. If $U$ is a nonempty subset of $V(G)$, then the subgraph $F$ of $G$ induced by $U$ is the graph such that $V(F) = U$ and $E(F)$ consists of the edges of $G$ incident with two nodes of $U$.

A graph $G$ is said to be recursive if and only if the smaller orders of $G$ are induced subgraphs of large order. A network has the unit-incrementability if its order can be expanded by one at a time. Let $d(u, v)$ denote the shortest distance between two distinct nodes $u$ and $v$ in a connected graph $G$. It is defined as the minimum of path lengths, in terms of the number of edges, between $u$ and $v$. The diameter of graph $G$, denoted as $diam(G)$, is the maximum distance among all pairs of nodes. Thus, $diam(G) = \max\{d(u, v) \mid u, v \in V(G)\}$. The average distance of a graph is the average distance among all pairs of nodes in the graph. The node-connectivity $\kappa(G)$ of a connected graph $G$ is defined as the minimum number of nodes that can be
removed from $G$, resulting in a disconnected graph. The edge-connectivity $\kappa_1(G)$ can be defined similarly. If $\phi_{\text{min}}(G)$ denotes the minimum degree of $G$, then it is known that $\kappa(G) \leq \kappa_1(G) \leq \phi_{\text{min}}(G)$. For graph-theoretic terminology used but not defined in this paper, readers may refer to [16].

For convenience, several notations are given as follows: $\phi_{\text{max}}(G)$ and $\phi_{\text{min}}(G)$ respectively indicate the maximum and minimum degrees of a graph $G$. The notation $v_\alpha$ means a node in a graph with label $\alpha$, where as $\text{DEG}(v_\alpha; \beta)$ denotes the degree of node $v_\alpha$ in a graph $G(\beta)$ consisting of $\beta$ nodes, where $0 \leq \alpha < \beta$.

Let $BR(i) = (b_k b_{k-1} \ldots b_1)$ be the binary representation of a non-negative integer $i$, and $BR_p(i)$ denotes the $p$th bit of $BR(i)$, where the least significant bit corresponds to $p = 1$. When there is no confusion, $BR(i)$ and $i$ will be used interchangeably. The following notations are also defined: $i^{[j]} = (b_k b_{k-1} \ldots \overline{b_j} \ldots b_1)$, in which the $j$th bit of $i$ is complemented, $\overline{i} = (\overline{b_k b_{k-1}} \ldots \overline{b_1})$, where all bits of $i$ are complemented, and $i^{(m)} = (b_k b_{k-1} \ldots \overline{b_m} \ldots \overline{b_1})$, such that the rightmost $m$ bits of $i$ are complemented.

2.2. Communication Models

For the performance evaluation of the communication topologies, two communication models are assumed: one-port communication and all-port communication [47]. In the one-port communication model, only one input/output (I/O) port or communication link per processor (node) can be activated at a time to send or receive data. In the all-port communication model, all I/O ports of a processor can be activated simultaneously.

Assume that the interconnection networks are packet switched and the communication links are of full duplex transmission (i.e., the links can transmit messages in both directions at the same time.) Let $\tau$ be the communication latency (start-up
time) for a packet of $B$ elements in each interprocessor communication. Let $M$ be the number of elements to be broadcasted and $t_e$ be the transfer time for an element. For analyzing the communication complexity, it is assumed that $t_e = 1$, $\tau = 0$, and $B = 1.$

2.3. Embedding Costs

The embedding problem of a guest graph $G = (V_G, E_G)$ onto a host graph $H = (V_H, E_H)$ is to find a pair of mapping functions, $\Phi$ and $\Psi$, such that $\Phi : V_G \rightarrow V_H$ is a mapping of their vertices, while $\Psi : E_G \rightarrow \{\text{paths in } H\}$ is a mapping from edges in $E_G$ to paths in $H$.

There are four metrics which measure the cost of an embedding. The dilation of an edge $e$ in $G$ is the length of the path $\Psi(e)$ in $H$. The dilation of $G$ in an embedding is the maximum dilation over all edges. The dilation of an embedding reflects the stretch of edges in $G$ through the mapping function $\Psi$, which is a lower bound on the communication delay. An embedding with a small value of dilation is preferable.

The edge-congestion (or congestion in short) of $G$ is the maximum number of edges in $G$ which are mapped by function $\Psi$ to a single edge in $H$. Intuitively, an embedding with a small value of edge-congestion would have less queuing delay. The load of $G$ is the maximum number of nodes in $G$ mapped by $\Phi$ to a single node in $H$. The expansion of $G$ is the ratio $\frac{|V_H|}{|V_G|}$.

In our study, the node-mapping function $\Phi$ is considered as one-to-one, i.e. the maximum load of embedding is one. An embedding of a guest graph which is the subgraph of host graph has dilation and edge-congestion equal to one. Since the embedding costs of dilation and expansion work as trade-off, we consider an embeddings
such that its dilation cost is one and expansion is minimum.

2.4. Embedding Guest Network Structures

Several guest graphs used for embedding into a given host are defined in the following.

A Hamiltonian cycle (or circuit) in a graph $G$ is a simple cycle containing all the vertices of $G$. A Hamiltonian path is a path containing all the vertices of $G$. A graph $G$ is defined to be Hamiltonian if it has a Hamiltonian cycle.

![Diagram](image)

Figure 2.1: (a) Two-rooted complete binary tree $TCBT(3)$. (b) Full-ringed tree $FRT(3)$.

A complete binary tree of height $h$, denoted as $CBT(h)$, is a binary tree with $2^h$ leaves at level $h$, in which the root is at level 0. Two-rooted complete binary tree [12] of height of $h$, $TCBT(h)$, is a $CBT(h)$ with the root replaced by a path of length two. For example, Figure 2.1(a) shows $TCBT(3)$.

The organization of multiple, monolithic microprocessors into a tree structure with extra interconnections, generally called X-tree, has been studied in [33]. The basic structure of X-tree is a complete binary tree made of general purpose micropro-
cessors called X-node. Several possible configurations of X-tree have been examined, for example, half-ringed tree, full-ringed tree, and so on. A full-ringed (binary) tree, \( FRT(h) \), is a complete binary tree \( CBT(h) \) with a ring connection at each level of the tree. Figure 2.1(b) shows \( FRT(3) \). A half-ringed tree of height \( h \) is a \( FRT(h) \) with the removal of the ring edges between two nodes having the same parent.

![Diagram](image)

**Figure 2.2:** (a) Hypertree \( HT(4) \). (b) The 4-dimensional tree machine \( TM(4) \).

A hypertree [43] of height \( h \), denoted as \( HT(h) \), is a complete binary tree \( CBT(h) \) with some additional links, called hyper-edges, at each level of tree. The hyper-edges are chosen from the \( n \)-cube connections. The root of \( HT(h) \) at level 0 is labeled as 1. In \( HT(h) \), each nonterminal node \( i \) has two children, the left child \( 2i \) and right child \( 2i + 1 \). The level number for each node is the number of binary digits after the most significant one in its address. Let the ray number \( r = z + 1 \), where \( z \) is the number of consecutive trailing zeros in the level number. The specific bit \( b \) which is affected by the \( n \)-cube connections at level \( m \) is determined by \( b = \frac{m}{2^r} + \frac{1}{2} \), where \( b \)
is the bit number counting from the left with the most significant one as bit zero. Figure 2.2(a) shows the hypertree \( HT(4) \) of height 4. In this figure, the dashed lines are hyper-edges chosen from the \( n \)-cube connections.

A tree machine of dimension \( n \), denoted as \( TM(n) \), is a graph which consists of two complete binary trees \( CBT(n) \)'s, called upper and lower trees, respectively, connected back to back along with the common leaves. Figure 2.2(b) shows the 4-dimensional tree machine \( TM(4) \).

2.5. Hypercube-family of Networks

For the purpose of comparison of our proposed networks, we need the following definitions related to hypercube-based networks.

![Figure 2.3](a) The incomplete hypercube \( IQ_3 \). (b) The 4-dimensional binary hypercube \( Q_4 \).]

An \( n \)-dimensional binary hypercube network is modeled as a graph \( Q_n = (V, E) \), where the node-set \( V = \{v_i \mid 0 \leq i < 2^n \} \) consists of non-negative labeled nodes, and an edge \((v_i, v_j)\) exists if and only if the Hamming distance between nodes \( v_i \) and \( v_j \) is equal to one, denoted as \( \rho(v_i, v_j) = 1 \). The Hamming distance between two nodes is
the number of bit positions in which their corresponding binary representations differ. More precisely, 

\[ \rho(v_i, v_j) = \sum_{p=1}^{[\log(j+1)]} [BR_p(i) \oplus BR_p(j)] \]

where \( \oplus \) indicates the bitwise Exclusive-OR operation, and \( 0 \leq i < j \). Figure 2.3(a) shows the 4-dimensional binary hypercube \( Q_4 \).

An incomplete hypercube \( IQ^k_\ell \) [48] consists of two "complete" hypercubes, \( Q_n \) (the front cube) and \( Q_k \) (the back cube), where \( 0 \leq k < n \). The \( 2^n + 2^k \) nodes of \( IQ^k_\ell \) are numbered from 0 to \( 2^n + 2^k - 1 \), in which the nodes labeled from 0 to \( 2^n - 1 \) belong to the front cube, while those labeled from \( 2^n \) to \( 2^n + 2^k - 1 \) belong to the back cube. An \( IQ^k_\ell \) can be decomposed into \( (2^{n-k}+1) \) number of \( k \)-dimensional subcubes \( Q^i_k \) for \( 0 \leq i \leq 2^{n-k} \). Each node in \( Q^i_k \) has the address \((i \star k)\) such that \( i = (b_{n+1} b_n \ldots b_{k+1}) \), where symbol \( b_j \in \{0,1\} \) for all \( j \) and \( \star \) denotes don't care symbol. Figure 2.3(b) shows the incomplete hypercube \( IQ^3_1 \).

The generalized incomplete hypercube \( GIQ(N) \) [72, 26] for \( N \geq 1 \), is composed of several complete hypercubes of different orders, and can be formally defined as follows. A generalized incomplete hypercube of order \( N > 1 \), denoted as \( GIQ(N) = (V, E) \), is an undirected, connected graph in which \( V = \{v_i \mid 0 \leq i \leq N-1\} \) is the set of labeled nodes. Let \( v_i \) and \( v_j \), for \( i < j \), be two nodes in \( V \), each being represented in \([\log(N+1)]\) bits. Then there exists an edge \( e = (v_i, v_j) \in E \), if and only if \( \rho(v_i, v_j) = 1 \). Clearly, network \( GIQ(N) \) is a binary hypercube \( Q_n \), when \( N = 2^n \) and is an incomplete hypercube \( IQ^\ell_n \), when \( N = 2^n + 2^k \). Thus, we are interested in the case when \( 2^{k-1} < N < 2^k \) and \( k > 1 \). Readers can refer to [26] for the discussion of structural properties of generalized incomplete hypercubes.

2.6. Enhanced Hypercube-family of Networks

The enhanced incomplete hypercube \( EIQ^\ell_k \) has extra edges added to \( IQ^\ell_k \). Let
Figure 2.4: (a) The enhanced incomplete hypercube $EIQ_1^3$. (b) The 4-dimensional folded hypercube $FQ_4$.

An $n$-dimensional folded hypercube, $FQ_n$, is designed by adding the complementary edges $(v_i, v_j)$ for all nodes $v_i$ in a hypercube $Q_n$. Note that the complementary edges in $FQ_n$ has Hamming distance of $n$. Clearly, $Q_n$ is a spanning subgraph of $FQ_n$. Later, we will see that $FQ_n$ is a spanning subgraph of the $n$-dimensional Hamming cube $HC_n$ consisting of $2^n$ nodes. Figure 2.4(b) shows the folded hypercube network $FQ_4$ in which the complementary edges are drawn by dashed lines.

### 2.7. Hypercube-like family of Networks

In an $n$-dimensional binary hypercube $Q_n$, let $C$ be a shortest cycle which is cycle of four vertices $(v_i, v_j, v_k, v_l)$. Edges $(v_i, v_j)$ and $(v_k, v_l)$ are two independent edges in $C$. The twisted $n$-cube $TQ_n$ corresponding to $Q_n$ is constructed by deleting $(v_i, v_j)$
and \((v_k, v_l)\) and adding two new twisted edges \((v_i, v_k)\) and \((v_j, v_l)\). It can be seen that a twisted edge has Hamming distance of two. Figure 2.5(a) shows a twisted 4-cube, \(TQ_4\). In this figure, the edges \((v_3, v_{11})\) and \((v_{15}, v_7)\) in \(C = (v_3, v_{11}, v_{15}, v_7)\) have been removed, and \((v_3, v_{15})\) and \((v_{11}, v_7)\) are the newly added twisted edges indicated by the dashed lines.

Let us now define crossed cubes. Two binary string with length of two are pair-related, denoted as \(x \sim y\), if and only if \((x, y) \in \{(00, 00), (10, 10), (01, 11), (11, 01)\}\). An \(n\)-dimensional crossed cube \(CC_n\) has the vertex-set \(V = \{v_i \mid 0 \leq i < 2^n\}\). When \(n = 1\), \(CC_1\) is the complete graph \(K_2\) of order two. For \(n > 1\), \(CC_n\) consists of two \((n - 1)\)-dimensional crossed cubes, \(CC_{n-1}^0\) and \(CC_{n-1}^1\) such that \(CC_{n-1}^0\) contains the nodes \(u = (0u_{n-2} \ldots u_0)\), while \(CC_{n-1}^1\) contains the nodes \(v = (0v_{n-2} \ldots v_0)\), where \(u_i, v_i \in \{0, 1\}\) for all \(i\) and \(j\). There exists an edge between nodes \(u\) and \(v\) if and only if \(u_{n-2} = v_{n-2}\) if \(n\) is even, and \(u_{2i+1} \neq v_{2i+1}\) for \(0 \leq i < \left\lfloor \frac{n-2}{2} \right\rfloor\). Since pair-related two-bit strings differ in at most one bit and there are \(\left\lfloor \frac{n-2}{2} \right\rfloor\) such pairs of substring in the labels of nodes \(u\) and \(v\), the Hamming distance between \(u\) and \(v\) is at most \(\left\lfloor \frac{n-2}{2} \right\rfloor + 2 = \left\lfloor \frac{n}{2} \right\rfloor + 1\). Furthermore, since crossed cubes are recursive networks, an
An n-dimensional crossed cube can have edges whose Hamming distances are from 1 up to \( \lfloor \frac{n}{2} \rfloor + 1 \). This result is important in the network classification by our model. Figure 2.5(b) shows the 4-dimensional crossed cube \( CC_4 \). The edges which are different from those of \( Q_4 \) are indicated by dashed lines.

![Diagram of 4-dimensional crossed cube](image)

**Figure 2.5:** (b) The 4-dimensional crossed cube \( CC_4 \).

Consider a network of order \( N = \sum_{i=0}^{q} 2^{p_i} \) such that \( p_{i+1} > p_i \) for \( 0 \leq i \leq q - 1 \). An incremental extensible hypercube \([70]\) \( IEQ(N) \) consists of \( q \) number of binary hypercubes \( Q_{p_i} \) of dimension \( p_i \) for \( 0 \leq i \leq q \). By using a different labeling mechanism, a node in subcube \( Q_{p_i} \) has the form \( (1^{n-p_i}a^{p_i+1}) \), where \( n = \lfloor \log N \rfloor \) and \( a \in \{0, 1\} \). Then \( IEQ(N) \) is iteratively constructed from the subcubes by defining the external edges among them. Let \( G^i \) be a composite graph, initially \( G^0 = Q_{p_0} \). For each iteration \( i \), for \( 0 \leq i < q \), a node \( (1^{n-p_i}b_{p_i}b_{p_i-1}...b_0) \) in \( G^i \) is connected to the node \( (1^{n-p_i-1}b_{p_i}b_{p_i-1}...b_0) \) of \( Q_{p_{i+1}} \) if \( p_{i+1} = p_i + 1 \); otherwise, it is connected to \( (p_{i+1} - p_i) \) different nodes of \( Q_{p_{i+1}} \), chosen by the way: \( (1^{n-p_{i+1}-1}1...1 b_{p_i}b_{p_i-1}...b_0) \), \( (1^{n-p_{i+1}-1}0...0 b_{p_i}b_{p_i-1}...b_0) \), and \( (1^{n-p_{i+1}-1}01...1 b_{p_i}b_{p_i-1}...b_0) \). Let \( G^{i+1} \) be \( G^i \cup Q_{p_{i+1}} \) and their corresponding external edges. Repeat to generate \( G^{i+2}, G^{i+3}, \)
..., and \( G^q \). The graph \( G^q \) resulting from the final iteration is the \( IEQ(N) \). Figure 2.6(a) shows network \( IEQ(11) \).

For a supercube \( SC(N) = (V, E) \) of order \( N \), each node is represented by a binary string of length \( n = \lfloor \log N \rfloor \). The vertex-set of \( SC(N) \) consists of nodes labeled from 0 to \( N - 1 \), and is divided into several subsets \( U = V_1 \cup V_2 \) and \( V_3 \). Sets \( U \) and \( V_3 \) contain the nodes having the most significant bits 0 and 1, respectively. Let a node \( v_j \) be the image of a node \( v_i \) if their binary string differ only in the most significant bit. Now \( V_1 \) is the set of nodes which are images of the nodes in \( V_3 \). There exists an edge \((v_i, v_j) \in E \) if and only if the Hamming distance \( \rho(v_i, v_j) = 1 \) or 2, for \( v_i \in V_2 \) and \( v_j \in V_3 \). Figure 2.6(b) shows an example of \( SC(11) \).

Note that when the network order is a power of two, many of the preceding networks such as incremental extensible hypercube, (generalized) incomplete hypercube, or supercube are nothing but complete binary hypercubes.
CHAPTER 3

THEORETICAL NETWORK MODEL

In this chapter, we first develop an algebraic group which forms the basis of our theoretical framework for network design. Based on the group theoretical properties, we then define the Hamming group model for generating interconnection networks, and later enhance it with the unit incrementability feature. The associated graphs are called Hamming group graphs and incremental Hamming graphs, respectively.

3.1. Hamming Group

In Group theory [42, 52], a group \((G, *)\) is a non-empty set \(G\) associated with a binary operation \(*\) satisfying the following axioms:

1. \(G\) is closed under \(*\): if \(a, b \in G\), then \(a * b \in G\).
2. \(*\) is associative in \(G\): for all \(a, b, c \in G\), \((a * b) * c = a * (b * c)\).
3. \(G\) has an identity element: there exists a special element \(e \in G\) such that \(e * a = a * e = a\) for all \(a \in G\). (\(e\) is also called unit element of \(G\).)
4. Existence of inverse: for every element \(a \in G\), there exists an element \(b \in G\) such that \(a * b = b * a = e\). Element \(b = a^{-1}\) is called the inverse of \(a\).

Let \(H = \{i | 0 \leq i < 2^n\}\) be a set of non-negative integers and \(\oplus\) be the bitwise Exclusive-OR operation. Then \((H, \oplus)\) is shown to be a group. This particular group will be called Hamming group.
Theorem 3.1. Let $\oplus$ denote the bitwise Exclusive-OR operation on the binary representation of the elements of the integer set $H = \{i \mid 0 \leq i < 2^n\}$. Then $(H, \oplus)$ is a group.

Proof: We prove that $(H, \oplus)$ satisfies the preceding four axioms in the definition of a group.

(1) Let $a, b \in H$, each being represented by a binary string of length $n$. Also, let $c = a \oplus b$. Since $0 < c < 2^n$, $c \in H$.

(2) The bitwise Exclusive-OR operation is an associative operation.

(3) Let $e = 0^n$ be an identity element in $H$. For all element $a \in H$, $e \oplus a = a = a \oplus e$.

(4) Since $a \oplus a = e$, the inverse of an element is the element itself, i.e. $a = a^{-1}$ for all $a \in H$.

It can be verified that the Hamming group $(H, \oplus)$ is a subgroup of $(G, \oplus)$, where $G$ is the set of all non-negative integers.

3.2. Hamming Group Graphs

A transformation graph $TG = (V, \Phi)$ is defined by a set $V$ of vertices and a set $\Phi$ of transformations of $V$. For each $v \in V$ and $\phi \in \Phi$, there is an arc labeled by $\phi$ from vertex $v$ to $\phi(v)$. The directed graph underlying the transformation graph $TG$ is obtained by erasing the labels from the arcs of $TG$ and removing any resulting parallel arcs.

A group-action graph $GAG = (V, \Pi)$ is a transformation graph for which each transformation in the set $\Pi$ is a permutation of the set $V$ [6]. Using the $GAG$, group graphs [3, 5] can be defined as follows: a group graph (or Cayley graph) is a
\(GAG = (V, \Pi),\) where \(V\) is the group \(Gr(\Pi)\) generated by \(\Pi\), and each \(\pi \in \Pi\) acts on \(Gr(\Pi)\) by right multiplication. For example, vertex \(v \cdot \pi\) results from multiplying \(v\) by \(\pi\). The transformation-set \(\Pi\) is called the set of generators of the group \(Gr(\Pi)\). Using these group-theoretical techniques, we can define graphs of corresponding to the Hamming groups.

A Hamming-group graph \(HGG = (V, \Omega)\) is also a transformation graph for which set \(V\) is the Hamming group having integers \(\{i \mid 0 \leq i < 2^n, \text{ and } n \geq 1\}\), i.e. all binary strings of length \(n\). A generator \(\omega \in \Omega\) is a binary string \(u\) of length \(n\) such that the image of vertex \(v\) transformed by \(\omega\) is \(\omega(v) = u \oplus v\), where \(\oplus\) is the bitwise Exclusive-OR operation.

By the action of a generator (Exclusive-OR operation), the image of a vertex \(v\) has possible \(1 \leq i \leq n\) bits complemented from \(v\). The number and positions of complemented bits depend on those bits which are 1's in the generator. Therefore, the \(2^n\) generators can be divided into \(n + 1\) subsets \(\Omega = \cup_{0 \leq i \leq n} \Omega_i\), where \(\cup\) is the union operation on sets. A generator \(\omega \in \Omega_i\) has \(i\) number of 1's bits in its binary representation, and there are \(\binom{n}{i}\) generators in \(\Omega_i\).

As an example, consider \(n = 3\). The eight generators can be divided into three subsets: \(\Omega_0 = \{000\}, \Omega_1 = \{001, 010, 100\}, \Omega_2 = \{011, 101, 110\},\) and \(\Omega_3 = \{111\}\). Let a generator \(\omega \in \Omega_i\). By the property of the bitwise Exclusive-OR operation, the Hamming distance between vertex \(v\) and its image \(\omega(v)\) is given by \(p(v, \omega(v)) = i\). Thus, the generators in set \(\Omega_i\) will be called the \(i\)-dimensional (Hamming-group) generators.

It is clear that the 0-dimensional generator \(\omega = 0^n\), the identity element of the Hamming group, transforms a vertex to itself, which results in a self-loop edge of Hamming-group graphs. Thus, this dimensional generator will be devoid of our
Figure 3.1: Hamming group graphs $HGG = (V, \Omega)$, where set $V = \{i \mid 0 \leq i < 2^n\}$ and (a) $\Omega = \Omega_1$, (b) $\Omega = \Omega_1 \cup \Omega_2$, (c) $\Omega = \Omega_1 \cup \Omega_3$, and (d) $\Omega = \Omega_1 \cup \Omega_2 \cup \Omega_3$.

discussion. Furthermore, since the generator set $\Omega$ is closed under inverse, Hamming-group graphs can be viewed as being undirected.

Figure 3.1 depicts examples of a few Hamming group graphs with different Hamming-group generators. The $n$-dimensional binary hypercube $Q_n$ is a Hamming group graph $HGG = (V, \Omega)$, where the set $\Omega = \Omega_1$ generates the edges with Hamming distance of one. Clearly, if $\Omega$ consists of all dimensional Hamming-group generators, $\Omega = \cup_{0<i\leq n} \Omega_i$, then $HGG = (V, \Omega)$ is a complete graph $K_{2^n}$ of order $2^n$.

Since we are interested in the supergraphs of the well-known binary hypercubes and some of their popular variants (Section 2.5—2.7), we assume that the generator set $\Omega$ for the Hamming group graphs under consideration is a superset of $\Omega_1$. For example, a twisted $n$-cube $TQ_n$ twists a pair of edges in a shortest cycle (consisting of four nodes) of $Q_n$. Since a twisted edge can simply be generated by $\Omega_2$, the network $TQ_n$ is a subgraph of $HGG = (V, \Omega)$ in which $\Omega = \Omega_1 \cup \Omega_2$. A crossed cube [35], whose design is based on recursively twisting the edges of $Q_n$, is also a subgraph of
Hamming group graphs with $\Omega = \Omega_1 \cup \Omega_2$.

3.3. Properties of Hamming Group Graphs

Due to the transformation of generators, Hamming-group graphs are undirected. Clearly, the underlying directed graph consists of the nodes whose indegrees and outdegrees are equal. Consider a Hamming-group graph $HGG = (V, \Omega)$, where generator set $\Omega = \cup_{i=1}^{j} \Omega_i$, for $0 < i \leq j$, $p_i \leq n$. The $HGG$ has node-degree equal to $\sum_{i=1}^{j} \binom{n}{p_i}$.

A graph is said to be vertex- or edge-symmetric if for every pair of vertices or edges, there exists an automorphism of the graph that maps one vertex or edge to another. The Hamming-group graphs are vertex-symmetric as proven in the following theorem. We provide a condition under which a Hamming graph is also edge-symmetric.

**Theorem 3.2.** The Hamming-group graphs are vertex-symmetric.

**Proof:** We need to show that given any two vertices $a, b \in V$ in the Hamming-group graph $HHG = (V, \Omega)$, there exists an automorphism on $V$ that maps $a$ to $b$. Consider the transformation on the Hamming group $b = \omega(a)$ for $\omega \in \Omega$. Assume that $b = \omega(a) = c \oplus a$. It implies that $c = b \oplus a^{-1} = b \oplus a$ by the property of the inverse element. Clearly, transformation $c$ maps $a$ into $b$, and is also an automorphism of the graph that maps an arbitrary vertex $x$ into $\omega(x) = c \oplus x$. $\square$

**Theorem 3.3.** The Hamming-group graphs are edge-symmetric if and only if the generators for the pair of edges are the same.

**Proof:** Consider two pairs of edges $(u, v)$ and $(u', v')$. Let $\omega$ and $\omega'$ be the generators of the edges such that $v = \omega(u) = c \oplus u$ and $v' = \omega'(u') = c' \oplus u'$, respectively. Assume that $u' = \delta(u) = d \oplus u$. We want to show that $v' = \delta(v)$. Since $u' = \delta(u) = d \oplus u$,
Figure 3.2: Incremental Hamming graphs $IHG(N)$ with (a) $\Omega = \Omega_1$ and (b) $\Omega = \Omega_1 \cup \Omega_2$, where $2 \leq N \leq 8$.

\[ d = u' \oplus u^{-1} = u' \oplus u. \]

Also, since $v = \omega(u) = c \oplus u$, $c = v \oplus u^{-1} = v \oplus u$. Therefore,

\[ \delta(v) = d \oplus v = c' \oplus c \oplus v' \] 

which is equal to $u'$ only if $c' \oplus c = e$. Thus, $c' = c^{-1} = c$. \( \square \)

3.4. Incremental Hamming Graphs

Because of the method of group generation, a group graph cannot expand its order by one at a time — such a property is called unit incrementability. In order to enhance a network with unit incrementability, we need to redefine the node-adjacency that includes a relationship between the newly added node and the existing ones. One possible method is outlined here. Let the generators $\omega \in \Omega$ precisely define the adjacency of a new node $v_{k+1}$ with those in the set $V = \{v_j \mid 0 \leq j \leq k\}$. Then a new version of $HHG = (V, \Omega)$, called the incremental Hamming graph, can grow its order by one. It is defined as follows.

An incremental Hamming graph of order $N \geq 2$, denoted as $IHG(N) = (V, E)$, has the node-set $V = \{v_i \mid 0 \leq i < N\}$ and the edge-set $E = \{(v_i, v_j) \mid v_j = \omega(v_i) \}$.
and $\omega \in \Omega_k$, where $0 \leq i < j < N$ and $1 \leq k \leq \lfloor \log N \rfloor$.

Figures 3.2(a) and 3.2(b) depict $IHG(N)$ with $\Omega = \Omega_1$ and $\Omega = \Omega_1 \cup \Omega_2$ for $2 \leq N \leq 8$, respectively. Figure 3.2(a) shows that a generalized incomplete hypercube $GIQ(N)$ is an incremental Hamming graph corresponding to $\Omega = \Omega_1$. Since an incomplete hypercube $IQ_k^N$ is a special case of $GIQ(N)$ with specific increment of $2^k$, $IQ_k^N$ fall into the same class of $GIQ(N)$. The node-set of a supercube [8] is divided into three subsets according to the most significant bit of node-labels, and there exists an edge between two nodes if their Hamming distance is 1 or 2. Thus, a supercube $SQ(N)$ is a subgraph of $IHG(N)$ with $\Omega = \Omega_1 \cup \Omega_2$. An incrementally extensible hypercube $IEQ(N)$ [70] has the edges with either Hamming distance 1 or 2, therefore, it also falls in this category. By carefully choosing the generators $\Omega_i$ that define the node adjacency, many other networks with unit incrementability can be derived from our proposed incremental Hamming graph model. This concept is important in our discussion.

3.5. Types of Hamming Generators

We can classify the set $\Omega$ of Hamming generators into two categories: invariant and variant. An invariant generator set comprises fixed number of Hamming-distance relations which can be explicitly enumerated, whereas in a variant generator set, the Hamming-distance relation is a function of some parameters of the networks, such as node-labels, network dimension, or network size (order), and so on. We denote the variant generator set as $\Omega_v = \{\omega \mid \pi \text{ is a function of network parameters}\}$. Figure 3.3 shows a schematic diagram of our network model, which categorizes several known families of interconnection networks.

It is mentioned in Section 2.6 that an $n$-dimensional folded hypercube is obtained
by adding the complementing edges \((u_i, v_i)\) for all nodes \(v_i\) in a hypercube \(Q_n\). These edges can be generated by the \(n\)-dimensional Hamming-group generators \(\Omega_n\). Thus folded hypercubes belong to the Hamming group graphs with \(\Omega = \Omega_1 \cup \Omega_x\), where \(\Omega_x = \{\omega \mid \pi \text{ is a function of the network dimension, } n\}\). By the definition in Section 2.6, the enhanced edges in an enhanced incomplete hypercube correspond to the generators \(\Omega_n\) and \(\Omega_{n+1}\). So, this network is classified as the incremental Hamming graph with the variant generator set. Another family of incremental Hamming graphs
with variant generators is the Hamming cubes (introduced in Chapter 4).

3.6. Summary

The Hamming group \((H, \oplus)\) in which \(H = \{i \mid 0 \leq i \leq 2^n - 1\}\) and \(\oplus\) is the bitwise Exclusive-OR operation is shown to be an algebraic group. The corresponding group graphs, called Hamming-group graphs, provide a model that can generate many symmetric supergraphs of binary hypercubes and satisfy the vertex-symmetry and conditional edge-symmetry. The enhancement of the Hamming-group graphs with unit incrementability feature can generate supergraphs of incomplete hypercubes.
CHAPTER 4

THE HAMMING CUBE NETWORKS

This chapter presents our first new family of networks, called Hamming cubes (HCs), which are derived from incremental Hamming graphs with a variant generator set. The formal definition and salient topological properties follow.

4.1. Network Definition

A Hamming cube of order (or size) $N$, denoted as $HC(N) = (V, E)$, is an undirected, connected graph in which $V = \{v_0, v_1, \ldots, v_{N-1} \mid v_i = BR(i)\}$ is the set of labeled nodes, where $BR(i)$ is the binary representation of $i$. If there is no confusion, we will use $BR(i)$ and $i$ interchangeably. Let $v_i$ and $v_j$, for $i < j$, be two nodes in set $V$, each being represented by $\lceil \log(j + 1) \rceil$ bits. Then there exists an edge $e = (v_i, v_j) \in E$, if and only if any one of the following two conditions is satisfied [30]:

(E1): The Hamming distance $\rho(v_i, v_j) = 1$; or

(E2): $\rho(v_i, v_j) = h = \lceil \log(j + 1) \rceil$ for $j \geq 1$.

The edges defined by Conditions $E1$ and $E2$ are designated as $E1$-edges and $E2$-edges, respectively. Clearly, the $E1$-edges define the underlying generalized incomplete hypercube topology of $HC(N)$, while the $E2$-edges are the additional enhanced edges. An $E2$-edge between two nodes $v_i$ and $v_j$ is said to be of dimension $n_h$ (or $n_h$-dimensional), if $\rho(v_i, v_j) = h = \lceil \log(j + 1) \rceil$ for $j \geq 1$. Note that $(0, 1)$ is an $E1$-edge as well as an $E2$-edge. Interpreted by our network model, $E1$- and $E2$-edges incident to $v_j$ can be generated by $1$- and $h$-Hamming-group generators, $\Omega_1$ and $\Omega_h$. 

31
respectively. Thus, a Hamming cube is an incremental Hamming graph with a variant generator set which is a function of node-labels.

Figures 4.1(a)—(h) depict Hamming cubes $HC(N)$ for $2 \leq N \leq 8$ and $N = 16$. The $E2$-edges are distinguished by broken lines. In the network $HC(16)$, since the Hamming distance between the nodes $v_3$ and $v_{12}$ is $\rho(v_3, v_{12}) = \lceil \log(12+1) \rceil = 4$, there exists an $n_4$-dimensional $E2$-edge between them. Similarly, $\rho(v_5, v_1) = 3$ implies that node $v_5$ is linked to $v_1$ through an $n_3$-dimensional $E2$-edge. For conformity, Hamming cube $HC(2^n)$ will be called the $n$-dimensional Hamming cube, denoted as $HC_n$. 
A binomial spanning tree of $HC(16)$, rooted at node $v_0$, using the $E2$-edges.

A binomial spanning tree\(^1\) rooted at node $v_0$ can be constructed by using the $E2$-edges as illustrated in Figure 4.2. This precisely gives one physical interpretation of the $E2$-edges in $HC_n$. The dimensions are labels on the edges. Note that in $HC_n$, there exists an $n$-dimensional $E2$-edge between all node-pairs $v_i$ and $v_j$ such that $\rho(v_i, v_j) = n$. Such edges correspond to the complementary edges in the folded hypercube ($FQ_n$) which is thus a spanning subgraph of $HC_n$.

By the definition of adjacency, Hamming cubes are recursive in nature and at least biconnected. The adjacency of a node can be determined on-the-fly by bit manipulation from its labeled address, and hence the networks can be succinctly represented. Contrary to the supercubes [64] or incrementally extensible hypercubes [70] of the same orders, Hamming cubes do not require reconfiguration of the existing network. Thus a Hamming cube of a large order can be constructed from small ones.

---

\(^1\)A binomial spanning tree is a binomial tree which spans all nodes in a network. Such a tree of height $n$ has the characteristic that the number of nodes at level $i$ is $\binom{n}{i}$ for $0 \leq i \leq n$. Also refer to Chapter 6 for details.
by adding extra edges to connect the smaller components appropriately.

4.2. Topological Properties

The following provides several structural properties of Hamming cubes which include such network metrics as edge complexity, node degree, and diameter.

**Property 4.1.** The number of edges in $HC_n$ is $E(N) = \frac{N}{2} \log N + N - 2$, where $N = 2^n$ and $n \geq 1$.

**Proof:** By definition, there are two types of edges in $HC_n$, namely the $E_1$- and $E_2$-edges. The $E_1$-edges correspond to those of the $n$-dimensional hypercube $Q_n$, having the total number $|E_1| = \frac{N}{2} \log N$, where $N = 2^n$ and $n \geq 1$. Because of the recursive definition, it can be proved by induction that the number of $E_2$-edges in $HC_n$ is $|E_2| = \sum_{i=1}^{n-1} 2^i = N - 2$. Thus, the total number of edges in $HC_n$ is

$$E(N) = |E_1| + |E_2| = \frac{N}{2} \log N + N - 2. \Box$$

The next property provides a general formula for computing the number of edges in a Hamming cube $HC(N)$ of arbitrary order $N \geq 2$.

**Property 4.2.** Let $N = 2^{p_q} + 2^{p_{q-1}} + \ldots + 2^{p_1}$ and $p_q > p_{q-1} > \ldots > p_1$.

Then $E(N) = E(2^{p_q}) + \sum_{j=1}^{q-1} [2^{p_j-1}(p_j + 2) + \sum_{i=1}^{j-1} 2^i]$ in $HC(N)$.

**Proof:** The node-set of $HC(N)$ can be partitioned into $q$ subsets, $V_q = \{v_0, v_1, \ldots, v_{2^{p_{q-1}}}\}$ and $V_l = \{v_j | \sum_{i=l+1}^{q} 2^i \leq j \leq (\sum_{i=1}^{q} 2^i) - 1\}$ for $1 \leq l \leq q - 1$. The subgraph of $HC(N)$ induced by $V_q$ forms an $HC_{p_q}$. Its edge complexity is given by Property 4.1. The induced subgraphs of the remaining subsets form $(q - 1)$ number of $p_j$-dimensional binary hypercubes for $1 \leq j \leq q - 1$, each having $p_j 2^{p_j-1}$ edges.

In addition, among all the induced subgraphs, there are $\sum_{j=1}^{q-1} \sum_{i=1}^{j} 2^i$ for $1 \leq
Let \( j \leq q - 1 \) number of \( E_1 \)-edges and \( \sum_{j=1}^{q-1} 2^{p_j} \) number of \( E_2 \)-edges. Thus, in total,

\[
E(N) = E(2^{p_1}) + \sum_{j=1}^{q-1} p_j 2^{p_j-1} + \sum_{i=1}^{j} \sum_{j=1}^{q-1} 2^{p_j} + \sum_{j=1}^{q-1} 2^{p_j} \\
= E(2^{p_1}) + \sum_{j=1}^{q-1} [2^{p_j-1}(p_j + 2) + \sum_{i=1}^{j} 2^{p_i}].
\]

**Example 4.1:** By Property 4.1, the total number of edges in networks \( HC_3 \) and \( HC_4 \) are \( E(8) = 18 \) and \( E(16) = 46 \), respectively. In network \( HC(13) \), \( N = 13 = (1101)_2 \) so that \( q = 3, p_3 = 3, p_2 = 2, \) and \( p_1 = 0 \). Hence by Property 4.2, the total number of edges is \( E'(13) = 33 \).

Let \( ONES[BR(i)] \) be the number of 1’s in \( BR(i) \). The definition of Hamming cubes immediately yields the degree of a node \( v_i \) in the Hamming cubes \( HC(i+1) \) as \( DEG(v_i; i+1) = ONES[BR(i)] + 1 \) for \( i > 1 \).

**Property 4.3.** The degree of a node \( v_i \) in \( HC_n \) is given by \( DEG(v_i; 2^n) = 2n - \lfloor \log(i + 1) \rfloor + 1 - \epsilon \), where \( n \geq 1 \) and

\[
\epsilon = \begin{cases} 
2 & \text{for } i = 0 \\
1 & \text{for } i = 1 \\
0 & \text{for } 2 \leq i \leq 2^n - 1.
\end{cases}
\]

**Proof:** Consider the label of a node \( v_i \) for \( 2 \leq i \leq 2^n - 1 \). Let the label \( BR(i) = (b_n b_{n-1} \ldots b_{p+1} b_p \ldots b_1) \), where \( p = \lfloor \log(i + 1) \rfloor \). Clearly, \( b_j = 0 \) for \( p < j \leq n \). By the \( E_1 \)-edges, there exists \( n \) nodes \( \{v_{i(d)}, v_{i(d-1)}, \ldots, v_{i(0)}\} \) adjacent to \( v_i \) at a Hamming distance of 1, where \( i^{[0]} = (b_k b_{k-1} \ldots b_j \ldots b_1) \). Moreover, according to the \( E_2 \)-edges, there are \( (n - \lfloor \log(i + 1) \rfloor + 1) \) nodes adjacent to \( v_i \) with Hamming distance \( h \) for \( p \leq h \leq n \). These nodes are \( \{v_{i(p)}, v_{i(p+1)}, \ldots, v_{i(n)}\} \).

Thus, the degree of \( v_i \) in \( HC_n \) is \( DEG(v_i; 2^n) = n + n - \lfloor \log(i + 1) \rfloor + 1 = 2n - \lfloor \log(i + 1) \rfloor + 1 \). In this analysis, nodes \( v_0 \) (or \( v_1 \)) has two (or one) adjacent nodes counted twice. Therefore, the parameter \( \epsilon \) is used to adjust the formula. \( \square \)
Clearly, a Hamming cube $HC(N)$ is regular with degree $N-1$ only for $N \leq 4$. By Property 4.3, the following corollary is immediate.

**Corollary 4.1.** The maximum degree of the $n$-dimensional Hamming cube $HC_n$ for $n \geq 3$, is $\phi_{\text{max}}(HC_n) = \text{DEG}(v_i; 2^n) = 2n - 1$ for $0 \leq i \leq 3$, and the minimum degree is $\phi_{\text{min}}(HC_n) = \text{DEG}(v_i; 2^n) = n + 1$ for $2^{n-1} \leq i \leq 2^n - 1$.

The next property gives the node-degree in a Hamming cube $HC(N)$ of arbitrary order.

**Property 4.4.** Consider the Hamming cube $HC(N)$, where $2^{k-1} < N < 2^k$ for $k > 1$. Let $N = \sum_{i=1}^{l} 2^{p_i}$, where $1 \leq l \leq k$ and $p_l > p_{l-1} > \ldots > p_1$. Also let $S_1 = \{\alpha \mid 0 \leq \alpha < 2^{p_1}\}$, $S_2 = \{\alpha \mid 0 \leq \alpha < \sum_{j=i+1}^{l-1} 2^{p_j}\}$, and $S_3 = \{\alpha \mid 2^{p_l} - \sum_{j=1}^{l-1} 2^{p_j} \leq \alpha < 2^n\}$. Then the degree of a node $v_\alpha$ in $HC(N)$ is given as

$$
\text{DEG}(v_\alpha; N) = \begin{cases} 
p_{(i-1)} + i + 2 & \text{for } \sum_{j=i-1}^{l-1} 2^{p_j} \leq \alpha < \sum_{j=i}^{l-1} 2^{p_j} + \sum_{j=1}^{l-1} 2^{p_j}; \\
p_{(i-1)} + i + 1 & \text{for } \sum_{j=i}^{l-1} 2^{p_j} + \sum_{j=1}^{l-1} 2^{p_j} \leq \alpha < \sum_{j=i-1}^{l-1} 2^{p_j}; \\
p_l + l & \text{for } \sum_{j=2}^{l} 2^{p_j} \leq \alpha < \sum_{j=1}^{l} 2^{p_j}; \\
\text{DEG}(v_\alpha; 2^{k-1}) + 2 & \text{for } S_2 \cap S_3; \\
\text{DEG}(v_\alpha; 2^{k-1}) + 1 & \text{for } (S_2 \cup S_3) - (S_2 \cap S_3); \\
\text{DEG}(v_\alpha; 2^{k-1}) & \text{for } S_1 - (S_2 \cup S_3),
\end{cases}
$$

where $1 \leq i \leq l - 2$.

**Proof:** We prove the degree of a node by counting its incident edges. Consider the above six cases, each for different nodes. Let $N = \sum_{i=1}^{l} 2^{p_i}$, where $1 \leq l \leq k$ and $p_{i+1} > p_i$. As will be shown in Section 4.3, Hamming cube $HC(N)$ can be decomposed into several induced subgraphs, say $HC(N) = \{HC_{p_1}, Q_{p_{l-1}}, \ldots, Q_{p_1}\}$, each induced by the node-subsets $V_{p_1} = \{v_\alpha \mid 0 \leq \alpha < 2^{p_1}\}$ and $V_{p_{(i-1)}} = \{v_\alpha \mid \sum_{j=i-1}^{l-1} 2^{p_j} \leq \alpha < \sum_{j=i-1}^{l-1} 2^{p_j}\}$ for $1 \leq i \leq l - 1$, respectively.

Cases 1 and 2 consider the nodes in the subgraphs $Q_{p_{(i-1)}}$ for $1 \leq i \leq l - 2$. Each
node has \((p_{l-i} + i)\) number of \(E1\)-edges and one \(E2\)-edge linked to its neighbors. The \(p_{l-i}\) number of \(E1\)-edges are internal edges linked to the other nodes in the hypercube \(Q_{p_{l-i}}\), while the \(i\) number of \(E1\)-edges are cross (external) edges, each linked to a node in subgraphs \(\{HC_{p_i}, Q_{p_{(l-i)_1}}, \ldots, Q_{p_{(l-i+1)}}\}\). The remaining one edge is an \(n_k\)-dimensional \(E2\)-edge linked to \(HC_{p_i}\). However, each node in Case 1 has one more \(E1\)-edge linked to a node in subgraphs \(\{Q_{p_{(l-i+1}_1}, Q_{p_{(l-i+2)}}}, \ldots, Q_l\}\).

Case 3 can be regarded as a special case of Cases 2. It considers the nodes in the subgraph \(Q_{p_i}\). Each node has \(p_i\) internal \(E1\)-edges, \((l - 1)\) cross edges, and one \(n_k\)-dimensional \(E2\)-edge. So, in total, there are \((p_i + l)\) edges.

The remaining cases – Cases 4, 5, and 6 – deal with the nodes in the subgraph \(HC_{p_i}\). In addition to the internal edges in \(HC_{p_i}\), each node in the set \(S_2\) has a \(k\)-dimensional \(E1\)-edge linked to the other subgraphs, while a node in the set \(S_3\) has an \(n_k\)-dimensional \(E2\)-edge. Therefore, the nodes in \((S_2 \cap S_3)\) have these two incident edges. □

**Example 4.2:** Consider networks \(HC(11)\) and \(HC(13)\). By Property 4.4,

\[
DEG(v_\alpha; 11) = \begin{cases} 
4 & \text{for } \alpha = 8; \\
3 & \text{for } \alpha = 9; \\
3 & \text{for } \alpha = 10; \\
DEG(v_\alpha; 2^3) + 1 & \text{for } \alpha \in \{0, 1, 2, 5, 6, 7\}; \\
DEG(v_\alpha; 2^3) & \text{for } \alpha \in \{3, 4\}
\end{cases}
\]
and

\[
DEG(v_\alpha; 13) = \begin{cases} 
5 & \text{for } \alpha = 8; \\
4 & \text{for } \alpha = 9, 10, 11; \\
3 & \text{for } \alpha = 12; \\
DEG(v_\alpha; 2^3) + 2 & \text{for } \alpha \in \{3, 4\}; \\
DEG(v_\alpha; 2^3) + 1 & \text{for } \alpha \in \{0, 1, 2, 5, 6, 7\}
\end{cases}
\]

Now let us analyze the diameter of Hamming cubes.

**Property 4.5.** The diameter of \(HC_n\) is \(\text{diam}(HC_n) \leq \lceil \frac{n}{2} \rceil\) for \(n \geq 1\).

**Proof:** In \(HC_n\), each node is represented by a binary string of length \(n\). Let \(v_i\) and \(v_j\) be two nodes. Clearly, the Hamming distance between them is \(\rho(v_i, v_j) \leq n\). However, due to the \(E2\)-edges, \(\text{diam}(HC_n) < n\). If \(\rho(v_i, v_j) \leq \lceil \frac{n}{2} \rceil\), the distance is \(d(v_i, v_j) \leq \lceil \frac{n}{2} \rceil\) with the help of \(E1\)-edges.

For the case \(\lceil \frac{n}{2} \rceil < \rho(v_i, v_j) \leq n\), we choose a node \(v_t\) such that \(\rho(v_i, v_t) = n\), that is \(v_t = v_i + 1\), where \(i^m = (b_k b_{k-1} \ldots b_m \ldots b_1)\). Then \(d(v_i, v_t) = 1\) through an \(E2\)-edge, and \(d(v_i, v_j) = n - \rho(v_t, v_j) \leq \lceil \frac{n}{2} \rceil\) through \(E1\)-edges. Since \(d(v_i, v_j) = d(v_i, v_t) + d(v_t, v_j) \leq \lceil \frac{n}{2} \rceil\), the distance between any two distinct nodes in \(HC_n\) is at most \(\lceil \frac{n}{2} \rceil\). \(\square\)

**Property 4.6.** For Hamming cube \(HC(N)\), where \(2^{k-1} < N < 2^k\) and \(k \geq 2\), the diameter is given by \(\text{diam}(HC(N)) \leq \lceil \frac{k-1}{2} \rceil + 1 = \lceil \log_2 N \rceil + 1\).

**Proof:** We divide the node-set \(V = \{v_i \mid 0 \leq i \leq N - 1\}\) of \(HC(N)\) into two subsets, \(V_1 = \{v_i \mid 0 \leq i \leq 2^{k-1} - 1\}\) and \(V_2 = \{v_i \mid 2^{k-1} \leq i \leq N - 1\}\). Let \(v_i\) and \(v_j\) be two nodes in \(V\). Now consider the following three cases:

**Case 1:** Both \(v_i\) and \(v_j\) are in \(V_1\).

By Property 4.5, the distance between them is \(d(v_i, v_j) \leq \lceil \frac{k-1}{2} \rceil\).

**Case 2:** \(v_i \in V_1\) and \(v_j \in V_2\).

Let node \(v_j' \in V_1\) be the node corresponding to \(v_j\) and linked by an \(E1\) (or \(E2\)-
edge, i.e. $v'_j = v_{j(n-1)}$ or $v_{j(n)}$. Again by Property 4.5, the distance $d(v_i, v'_j) \leq \lceil \frac{k-1}{2} \rceil$. Therefore, $d(v_i, v_j) \leq \lceil \frac{k-1}{2} \rceil + 1$.

**Case 3:** Both $v_i$ and $v_j$ are in $V_2$.

In this case, the Hamming distance $\rho(v_i, v_j) \leq k - 1$. Now if $\rho(v_i, v_j) \leq \lceil \frac{k-1}{2} \rceil$, the distance between the given nodes is $d(v_i, v_j) \leq \lceil \frac{k-1}{2} \rceil$ through $E_1$-edges. On the other hand, if $\lceil \frac{k-1}{2} \rceil < \rho(v_i, v_j) \leq k - 1$, we consider a node $v'_j = v_{j(n)} \in V_1$. Since $d(v_j, v'_j) = 1$ through an $E_2$-edge and $d(v_i, v'_j) = k - \rho(v_i, v_j) \leq \lceil \frac{k-1}{2} \rceil$ through $E_1$-edges, we have $d(v_i, v_j) = d(v_i, v'_j) + d(v'_j, v_j) \leq \lceil \frac{k-1}{2} \rceil + 1$.

Analyzing Cases 1, 2 and 3, we obtain $\text{diam}(HC(N)) \leq \lceil \frac{\log N}{2} \rceil + 1$. \hfill \square

### 4.3. Network Comparison

Table 4.1 provides a comparison of the topological properties of Hamming cubes with several other hypercube-like networks. Note that $n$-dimensional binary hypercubes ($Q_n$), folded hypercubes ($FQ_n$) [36], twisted $n$-cubes ($TC_n$) [37], crossed cubes ($CC_n$) [35], and bridged hypercubes ($BQ_n$) [22] all grow in powers of two (i.e. $N = 2^n$). However, the generalized incomplete hypercubes $GIQ(N)$, incrementally extensible hypercubes $IEQ(N)$, supercubes $SC(N)$, enhanced generalized incomplete hypercubes $EGIQ(N)$ introduced in Chapter 5, and Hamming cubes $HC(N)$ grow with incrementability of one.

Interestingly, the proposed Hamming cube network $HC(N)$ of an arbitrary order $N$ has the diameter almost the same as the closest sized networks $CC_n$, $FQ_n$, or $BQ_n$, at the cost of at most $N - 2$ extra edges compared to networks $Q_n$, $TC_n$, or $CC_n$. Network $FQ_n$ has $\frac{N}{2}$ extra edges over $Q_n$ and bridge hypercube $BQ_n$ has $\left(\lceil \frac{\log N}{2} \rceil + 1 \right)$ additional edges over $Q_n$. Furthermore, the diameter of $HC(N)$ is the smallest among all the three unit-incremental networks. The fact that Hamming cubes are recursive
Table 4.1: Comparison of several hypercube-like networks.

<table>
<thead>
<tr>
<th>Networks of $N$ nodes</th>
<th>network expansion</th>
<th># edges $E(N)$ for $N = 2^n$</th>
<th>degree ($\phi$) for $N \geq 2$</th>
<th>regular? / reconfiguration required?</th>
<th>diameter for $N \geq 2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binary Hypercube $Q(N)$</td>
<td>$N$</td>
<td>$\frac{N}{2} \log N$</td>
<td>$\log N$</td>
<td>yes/no</td>
<td>$\log N$</td>
</tr>
<tr>
<td>Twisted hypercube $TQ(N)$</td>
<td>$N$</td>
<td>$\frac{N}{2} \log N$</td>
<td>$\log N$</td>
<td>yes/yes</td>
<td>$\log N \sim 1$</td>
</tr>
<tr>
<td>Folded hypercube $FQ(N)$</td>
<td>$N$</td>
<td>$\frac{N}{2} (\log N + 1)$</td>
<td>$\log N + 1$</td>
<td>yes/yes</td>
<td>$\lfloor \log N \rfloor + 1$</td>
</tr>
<tr>
<td>Crossed cube $CC(N)$</td>
<td>$N$</td>
<td>$\frac{N}{2} \log N$</td>
<td>$\log N$</td>
<td>yes/no</td>
<td>$\lfloor \log N \rfloor + 1$</td>
</tr>
<tr>
<td>Bridged hypercube $BQ(N)$</td>
<td>$N$</td>
<td>$\frac{N}{2} \log N + \epsilon$</td>
<td>$\log N \leq \phi \leq \log N + \epsilon$</td>
<td>no/no</td>
<td>$\lfloor \log N \rfloor + 1$</td>
</tr>
<tr>
<td>Enhanced Incomplete Hypercube $EIQ(N)$</td>
<td>$2^k$</td>
<td>$\frac{N}{2} (\log N + 1)$</td>
<td>$2 \leq \phi \leq \log N + 1$</td>
<td>no/yes</td>
<td>$\lfloor \log N \rfloor + 1$</td>
</tr>
<tr>
<td>General Incomplete Hypercube $GIQ(N)$</td>
<td>$1$</td>
<td>$\frac{N}{2} \log N$</td>
<td>$1 \leq \phi \leq \log N$</td>
<td>no/no</td>
<td>$\log N$</td>
</tr>
<tr>
<td>Enhanced Generalized Incomplete hypercube $EGIQ(N)$</td>
<td>$1$</td>
<td>$\frac{N}{2} (\log N + 1)$</td>
<td>$\log N + 1$</td>
<td>no/yes</td>
<td>$\lfloor \log N \rfloor + 1$</td>
</tr>
<tr>
<td>Supercube $SC(N)$</td>
<td>$1$</td>
<td>$O(N \log N)$</td>
<td>$2 \leq \phi \leq 2 \log N - 2$</td>
<td>no/yes</td>
<td>$\log N$</td>
</tr>
<tr>
<td>Incrementally Extensible Hypercubes $IEQ(N)$</td>
<td>$1$</td>
<td>$\frac{N}{2} \log N$</td>
<td>$\log N \leq \phi \leq \log N + 1$</td>
<td>no/yes</td>
<td>$\lfloor \log N \rfloor + 1$</td>
</tr>
<tr>
<td>Hamming cube $HC(N)$</td>
<td>$1$</td>
<td>$\frac{N}{2} \log N + N - 2$</td>
<td>$3 \leq \phi \leq 2 \log N - 1$</td>
<td>no/no</td>
<td>$\lfloor \log N \rfloor + 1$</td>
</tr>
</tbody>
</table>

\[ \epsilon = \left( \frac{\log N}{\log N + 1} \right) + 1 \]

in nature (that is the smaller order Hamming cubes are induced subgraphs of a large order), implies that they do not require reconfiguration while expanding, as opposed to incrementally extensible hypercubes or supercubes.

Table 4.2 shows the comparison of $IQ^k_n$ and Hamming cube $HC(2^n + 2^k)$ in terms of node degree, edge complexity, and diameter. The diameter of $HC(2^n + 2^k)$ is approximately half of $IQ^k_n$ in trade-off of a larger range of node-degree. However, their edge-complexity are very close. When $k = n - 1$, the $IQ^n_{n-1}$ has only two type of degrees, $n$, and $n - 1$, shown as in Table 4.3. For the same case, $HC(2^n + 2^{n-1})$
Table 4.2: Comparison of \( IQ^p_k \) and \( HC(2^n + 2^k) \)

<table>
<thead>
<tr>
<th>networks</th>
<th>( IQ^p_k )</th>
<th>( HC(2^n + 2^k) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>degree</td>
<td>([n - k + 1, n + 1])</td>
<td>([n - k + 2, 2n + 1])</td>
</tr>
<tr>
<td>edge complexity</td>
<td>( n2^{n-1} + (n - k + 2)2^{n-k-1} )</td>
<td>((n + 2)2^{n-1} + (n - k + 6)2^{n-k-1} - 4)</td>
</tr>
<tr>
<td>diameter</td>
<td>( n + 1 )</td>
<td>( \lceil \frac{n}{2} \rceil + 1 )</td>
</tr>
</tbody>
</table>

Table 4.3: Comparison of \( IQ^n_{n-1} \) and \( HC(2^n + 2^{n-1}) \)

<table>
<thead>
<tr>
<th>networks</th>
<th>( IQ^n_{n-1} )</th>
<th>( HC(2^n + 2^{n-1}) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>degree</td>
<td>([n, n + 1])</td>
<td>([n + 1, 2n + 1])</td>
</tr>
<tr>
<td>edge complexity</td>
<td>((3n + 1)2^{n-2})</td>
<td>((3n + 9)2^{n-2} - 4)</td>
</tr>
<tr>
<td>diameter</td>
<td>( n + 1 )</td>
<td>( \lceil \frac{n}{2} \rceil + 1 )</td>
</tr>
</tbody>
</table>

can have smaller range of node degree.

4.4. Decomposition of Hamming cubes

Since a Hamming cube is defined recursively from its smaller orders, the \( n \)-dimensional network \( HC_n \) can be decomposed into two induced (disjoint) subgraphs, denoted as \( HC_n = \{ HC_{n-1}, Q_{n-1} \} \). Here \( HC_{n-1} = (V', E') \) is the \((n - 1)\)-dimensional Hamming cube with the node-set \( V' = \{ v_\alpha \mid 0 \leq \alpha < 2^{n-1} \} \) and \( Q_{n-1} = (V'', E'') \) is the \((n - 1)\)-dimensional hypercube with the node-set \( V'' = \{ v_\alpha \mid 2^{n-1} \leq \alpha < 2^n \} \). We can further decompose the subgraph \( HC_{n-1} \) as \( \{ HC_{n-2}, Q_{n-2} \} \), and so on. Therefore, \( HC_n \) can be iteratively decomposed into \((n - 1)\) induced (disjoint) subgraphs. We will denote such a decomposition as \( \{ HC_2, Q_2, Q_3, \ldots, Q_{n-1} \} \). Note that subgraph \( HC_2 \) consisting of \( V' = \{ v_0, v_1, v_2, v_3 \} \) forms a complete graph \( K_4 \). The node-sets of other subgraphs are given by \( V^i = \{ v_\alpha \mid 2^i \leq \alpha < 2^{i+1} \} \), for \( 2 \leq i \leq n - 1 \).

Let \( HC_n = \{ HC_{n-2}, Q_{n-2}, Q_{n-1} \} \). By definition, each node \( v_\alpha \) in the subgraph
Figure 4.3: (a) Decomposition of $HC_3 = HC(8)$ with relabeling of nodes. (b) Decomposition of $HC(12)$ with node relabeling. (c) A decomposition of $HC(15)$.

$Q_{n-1}$ has two edges — one $(n-1)$-dimensional $E1$-edge linked to the node $v_{a^{(n)}}$ in network $HC_{n-2}$, and another $n$-dimensional $E2$-edge linked to node $v_{a^{(n)}}$ in the subcube $Q_{n-2}$. Therefore, we can relabel the node-set $V$ as $\{v_0, v_1, \ldots, v_{(2^n-2)}, v_{(2^n-1)}, v_{(2^n-2)}, \ldots, v_{2^n-1}\}$ such that the relabeled network is also an $n$-dimensional Hamming cube $HC_n$. Figure 4.3(a) shows the relabeled network $HC_3$, in which the solid lines are $E1$-edges, and the broken lines correspond to the $E2$-edges.

Due to this relabeling mechanism, the network $HC(2^n + 2^{n-1})$ can now be decomposed as $HC(2^n + 2^{n-1}) = \{HC_{n-1}, Q_{n-1}, Q_{n-1}\}$, and any two of the components form a hypercube $Q_n$. Figure 4.3(b) shows $HC(12) = \{HC_2, Q_2, Q_2\}$ with relabeling.

Considering $HC_n = \{HC_2, Q_2, Q_3, \ldots, Q_{n-1}\}$, the nodes in each induced subgraph have the same degree, and satisfy the node-symmetry. The following property is immediate from the above discussion and with the help of Property 4.3.

**Property 4.7.** In the $n$-dimensional Hamming cube $HC_n$ for $n > 1$, the degree of
nodes in each induced subgraph $HC_j$ is given by $\text{DEG}(v_i; 2^n) = n + j$, where

$$\begin{cases} 
1 \leq j \leq n - 2 & \text{for } 2^{n-j} \leq i \leq 2^{n-j+1} - 1, \\
 j = n - 1 & \text{for } 0 \leq i \leq 3.
\end{cases}$$

The recursive decomposition of the $n$-dimensional Hamming cube yields: $HC_n = \{HC_2, Q_2, Q_3, \ldots, Q_{n-1}\} = \{HC_3, Q_3, \ldots, Q_{n-1}\} = \{HC_4, Q_4, \ldots, Q_{n-1}\} = \ldots$ With the help of Property 4.7, the edges by which a node in the induced subgraph $HC_j$, where $3 \leq j \leq n - 1$, is connected to its neighbors can be characterized as follows.

**Property 4.8.** A node in the induced subgraph $HC_j$ of $HC_n$, where $3 \leq j \leq n - 1$, is connected to its neighbors by the edges in dimensions $\{0, 1, \ldots, j - 1, n_j, j, n_j + 1, \ldots, n - 1, n_n\}$.

**Example 4.3:** In the Hamming cube $HC_4$, consider node $v_0$ in $HC_2$ and $v_4$ in $HC_3$. Now the nodes $v_0$ and $v_4$ have the incident edges in dimensions $\{0, 1, 2, 2, n_3, 3, n_4\}$, and $\{0, 1, 2, n_3, 3, n_4\}$, respectively.

In a generalized incomplete hypercube $GIQ(N)$ or Hamming cube $HC(N)$, where $2^{k-1} < N < 2^k$ and $k > 1$, we can partition the node-set into several subsets. Let $N = \sum_{i=1}^{l} 2^{p_i}$, where $1 \leq l \leq k$ and $p_{i+1} > p_i$. The $l$ number of node-subsets are $V^{p_i} = \{v_\alpha \mid 0 \leq \alpha < 2^{p_i}\}$ and $V^{p_{i+1}} = \{v_\alpha \mid \sum_{j=i-l+1}^{l} 2^{p_j} \leq \alpha < \sum_{j=i}^{l} 2^{p_j}\}$ for $1 \leq i \leq l - 1$. For network $GIQ(N)$, each node-subset and its corresponding induced edges form a hypercube, denoted as $GIQ(N) = \{Q_{p_1}, Q_{p_2}, \ldots, Q_{p_l}\}$.

For network $HC(N)$, the only difference is that the node-subset $V^{p_i}$ forms a $p_i$-dimensional Hamming cube instead of a hypercube $Q_{p_i}$, that is $HC(N) = \{HC_{p_1}, Q_{p_1}, \ldots, Q_{p_l}\}$. Figure 4.3(c) shows the decomposition of $HC(15)$, and $HC(15) = \{HC_3, Q_2, Q_1, Q_0\}$. Generally, we can view a Hamming cube $HC(N)$ as a meta-union of a Hamming cube $HC_{k-1}$ and a generalized incomplete hypercube $GIQ(N - 2^{k-1})$.
denoted as $HC_{k-1} \odot GIQ(N - 2^{k-1})$. The meta-union operation on the nodes of $GIQ(N - 2^{k-1})$ satisfies conditions $E1$ and $E2$ of the proposed network definition.

4.5. Simple Routing Schemes

In this section, we provide two simple routing (one-to-one communication) schemes for the Hamming cubes. They are defined according to the proofs of Properties 4.5 and 4.6. Let $v_i$ and $v_j$ be two nodes in a Hamming cube, and $P = \{p_\alpha \mid 1 \leq \alpha \leq h\}$ be the set of bit positions in which labels of $v_i$ and $v_j$ differ. To recall the notations $i^{\bar{\alpha}}$ and $\bar{i}$, refers to Section 2.1.

Routing Scheme $RS_{HC_n}$: Route node $v_i$ to $v_j$ in $HC_n$, where $n \geq 1$.

1. Compute the Hamming distance $\rho(v_i, v_j) = h$.

2. If $(h \leq \lfloor \frac{h}{2} \rfloor)$ then

   begin
   2.1 Compute the set $P$ for $v_i$ and $v_j$.
   2.2 Let $t_\alpha$ be the number complementing the $(p_\alpha)$th bit of integer $i$,
   i.e. $t_\alpha = i^{\bar{p}_\alpha}$, where $p_\alpha \in P$ and $1 \leq \alpha \leq h$.
   2.3 Route node $v_i$ to $v_j$ along the path $(v_i, v_t, v_h, \ldots, v_{h-l}, v_j)$
   end

3. else begin

3.1 Compute the set $P$ for nodes $v_i$ and $v_j$.
3.2 Compute $t_\alpha$ as in Line 2.2.
3.3 Route node $v_i$ to $v_j$ along the path $(v_i, v_t, \ldots, v_{h-1}, v_j, v_j)$.
end

Next is the routing scheme for Hamming cube $HC(N)$, where $2^{k-1} < N < 2^k$ and $k \geq 2$. Divide the node-set $V = \{v_i \mid 0 \leq i \leq N - 1\}$ into two subsets, $V_1 = \{v_i \mid 0 \leq i \leq N - 2^{k-1}\}$ and $V_2 = \{v_i \mid N - 2^{k-1} < i \leq N - 1\}$.
i \leq 2^{k-1} - 1} and \( V_2 = \{ v_i \mid 2^{k-1} \leq i \leq N - 1 \} \). Then we consider three different cases according to the membership of nodes \( v_i \) and \( v_j \) in \( V_1 \) and \( V_2 \). Note that the subgraph induced by \( V_1 \) forms a Hamming cube \( HC_{k-1} \). Also, a node in \( V_1 \) has 0 in its most significant bit, that is \( BR_k(i) = 0 \).

Routing Scheme \( RS_{HC}(N) \): Route node \( v_i \) to \( v_j \) in \( HC(N) \), where \( 2^{k-1} < N < 2^k \)

1  If \( (BR_k(i) = 0) \) then

   begin

   1.1  If \( (BR_k(j) = 0) \) then

       Since \( v_i \) and \( v_j \) are in \( HC_{k-1} \), use Routing Scheme \( RS_{HC_n} \).

       else begin \( \{BR_k(j) = 1\} \)

       1.2.1 Consider two nodes \( v_{j[u]} \) and \( v_j \), and compute \( \rho(v_i, v_{j[u]}) = h' \) and \( \rho(v_i, v_j) = \overline{h} \).

       1.2.2 Without loss of generality, assume that \( h' \geq \overline{h} \). We first route \( v_i \) to \( v_{\overline{h}} \) by

       Routing Scheme \( RS_{HC_n} \), and then route \( v_{\overline{h}} \) to \( v_j \).

       1.2.3 For the case of \( h' < \overline{h} \), apply the same method but replace node \( v_{\overline{h}} \) with \( v_{h'} \).

       end

   end

2  else begin

2.1  if \( (BR_k(j) = 0) \) then use the reverse routing path of the case when

\( BR_k(i) = 0 \) and \( BR_k(j) = 1 \) as in Lines 1.2.1-1.2.3.

   else begin

2.2.1 Compute the Hamming distance \( \rho(v_i, v_j) = h \).

2.2.2 If \( (h \leq \lfloor \frac{k-1}{2} \rfloor) \) then Repeat Lines 2.1-2.3 of Routing Scheme \( RS_{HC_n} \).

2.2.3 else Repeat Lines 3.1-3.3 of Routing Scheme \( RS_{HC_n} \).

   end

end
Example 4.4: Consider nodes $v_9$, $v_3$, and $v_7$ in the Hamming cube $HC_4$. By Routing Scheme $RS_{HC_n}$, since $\rho(v_9, v_3) = 2$, the routing path from node $v_9$ to $v_3$ is $(v_9, v_{11}, v_3)$. Also, since $\rho(v_9, v_7) = 3$, the path routing node $v_9$ to $v_7$ is $(v_9, v_6, v_7)$. Now consider $HC(28)$ and its four nodes $v_4$, $v_{26}$, $v_{23}$, and $v_{24}$. By Routing Scheme $RS_{HC(N)}$, the paths from node $v_4$ to $v_{26}$ and from $v_{23}$ to $v_{24}$ are $(v_4, v_5, v_{26})$ and $(v_{23}, v_7, v_{24})$, respectively.

Since the routing path between any two nodes is determined by their labeled addresses, the routing schemes for Hamming cubes are self-routing.

4.6. Fault-tolerance and Reliability Analyses

In this section, we will analyze the fault-tolerance and reliability of Hamming cubes. We first show that a Hamming cube is optimally fault-tolerant since its node-connectivity is equal to the minimum degree.

4.6.1. Optimal Fault-tolerance

A graph is $\tau$-connected if $\kappa \geq \tau$. A theorem, originally due to Whitney [79], states that a graph is $\tau$-connected if and only if there are at least $\tau$ node-disjoint paths between every pair of nodes. We use this fact to prove the optimality of fault tolerance of Hamming cubes.

For convenience in our proofs of this section, we will denote the $n_h$-dimensional $E2$-edges, where $h = \lceil \log(N + 1) \rceil$, as the $d$-dimensional edges. Thus in the $n$-dimensional Hamming cube $HC_n$, there are at least $(n + 1)$ types of dimensional edges, designated as $D = \{d_1, d_2, \ldots, d_n, \overline{d}\}$, where edges with label $d_j$ for $1 \leq j \leq n$, are dimensions of $E1$-edges. According to Corollary 4.1, the minimum node-degree $\phi_{min}(HC_n) = n + 1$, for $n > 1$. Therefore, we have the following lemma.
Lemma 4.1. The \( n \)-dimensional Hamming cube \( HC_n \) is \((n+1)\)-connected for \( n > 1 \). Therefore, the node connectivity is \( \kappa(HC_n) = n + 1 = \phi_{\min}(HC_n) \), which is also equal to edge-connectivity.

Proof: As mentioned above, let the \((n + 1)\) types of dimensional edges in Hamming cube \( HC_n \) be \( D = \{d_1, d_2, \ldots, d_n, \overline{d}\} \). Assume that \( v_i \) and \( v_j \) be two distinct nodes in the network such that their Hamming distance is \( \rho(v_i, v_j) = l \), where \( 1 \leq l \leq n \). The bits in which \( v_i \) and \( v_j \) differ determine a subset \( D_1 \subseteq D \). Let \( D_1 = \{d'_1, d'_2, \ldots, d'_l\} \). Using these edges, a disjoint path \( (d'_1, d'_2, \ldots, d'_l) \) of length \( l \) between \( v_i \) and \( v_j \) can be easily obtained. In addition, cyclically shifting (to the right) of the dimensions in the path forms other \( l - 1 \) number of disjoint paths of length \( l \), designated as \( (d'_2, d'_3, \ldots, d'_l) \), \( (d'_3, d'_4, \ldots, d'_l) \), \ldots, and \( (d'_I, d'_I, \ldots, d'_{I - 1}) \).

Let \( D_2 = D - D_1 \). The cardinality of the set \( D_2 \) is \( |D_2| = n + 1 - l \). With a similar permutation approach, the edges in \( D_2 \) form another group of \( n + 1 - l \) disjoint paths of length \( n + 1 - l \). Since we use \((n + 1)\) distinct dimensional edges and the cyclic-shift-right operation to form the paths, no internal nodes in these paths will be the same. Therefore, these \((n + 1)\) number of paths are disjoint. □

Example 4.5: In \( HC_4 \), the Hamming distance \( \rho(v_6, v_8) = 3 \), \( D_1 = \{d_2, d_3, d_4\} \), and \( D_2 = \{d_1, \overline{d}\} \). There are five disjoint paths between nodes \( v_6 \) and \( v_8 \), given as \( \text{path}_1 = (v_6, v_4, v_0, v_8) \), \( \text{path}_2 = (v_6, v_2, v_{10}, v_8) \), \( \text{path}_3 = (v_6, v_{14}, v_{12}, v_8) \), \( \text{path}_4 = (v_6, v_7, v_8) \), and \( \text{path}_5 = (v_6, v_9, v_8) \).

Next, we provide several lemmas that are used to prove the node-connectivity for \( HC(N) \), \( 2^{k-1} < N < 2^k \), is also equal to its minimum degree. Lemma 4.2 related to the binary hypercube is provided for the sake of completeness.

Lemma 4.2. Let \( s \) be a node in the \( n \)-dimensional hypercube \( Q_n \). There are at most \( n \) other distinct nodes in \( Q_n \) such that the paths from \( s \) to them are disjoint.
Lemma 4.3. Let $s$ be a node in the $n$-dimensional Hamming cube $HC_n$ where $n > 1$. There exist at most $(n + 1)$ other distinct nodes in $HC_n$ such that the paths from $s$ to these nodes are disjoint.

Proof: (by induction) For the basis case $n = 2$, there are three types of dimensional edges in Hamming cube $HC_2$, given as $D = \{d_1, d_2, \overline{d}\}$, due to which three disjoint paths exist from $s$ to the other three nodes.

For the induction hypothesis, assume that the lemma is true for $n = 2^k$. To prove true for $n = 2^{k+1}$, we partition Hamming cube $HC_{k+1} = (V, E)$ into two subgraphs $HC_k = (V_1, E_1)$ and $Q_k = (V_2, E_2)$ by our decomposition, where $V_1 = \{v_i \mid 0 \leq i < 2^k\}$ and $V_2 = \{v_j \mid 2^k \leq i < 2^{k+1}\}$. The $(k + 2)$ types of dimensional edges are $D = \{d_1, d_2, \ldots, d_{k+1}, \overline{d}\}$. Let us consider the following two cases depending on the membership of node $s$.

Case 1: node $s$ in the subgraph $HC_k$, i.e. $s \in V_1$.

Assume that $l$ of $(k + 2)$ nodes are in $HC_k$. If $l = k + 1$, then by our induction hypothesis, there are $(k + 1)$ disjoint paths routing from node $s$ to these $(k + 1)$ nodes in $HC_k$. For another node $v_\alpha$ in $Q_k$, there is a new disjoint path formed with the help of either the $d_{k+1}$-dimensional edge $(s, s^{[k+1]}, \ldots, v_\alpha)$ or the $\overline{d}$-dimensional edge $(s, s^{(k+1)}, \ldots, v_\alpha)$.

If $l = k + 2$ for $k > 2$, all nodes will be in $HC_k$. By our induction hypothesis, we have one node $v_\alpha$ in $HC_k$ that cannot be reached from node $s$ by a disjoint path. With the help of the $d_{k+1}$- (or $\overline{d}$)-dimensional edges, the required path is $(s, s^{[k+1]}(or \, s^{(k+1)}), \ldots, v_\alpha^{(k+1)}(or \, v_\alpha^{(k+1)}), v_\alpha)$.

Now consider the case when $0 \leq l = m < k + 1$. Let $V'_1 = \{v_\alpha_1, v_\alpha_2, \ldots, v_\alpha_m\} \subseteq V_1$ and $V'_2 = \{v_\beta_1, v_\beta_2, \ldots, v_{\beta_{k+2-m}}\} \subseteq V_2$. Clearly, there are $m$ disjoint paths from node $s$ to $v_\alpha_i \in V'_1$. The following gives the $(k + 1 - m)$ disjoint paths from node
s to \((k + 1 - m)\) nodes in \(V_2\). We route \(s\) to the \((k + 1 - m)\) nodes belonging to 
\((V_1 - V'_1 - \{s\})\), indicated as \(V''_1 = \{v_{\gamma_1}, v_{\gamma_2}, \ldots, v_{\gamma_{k+1-m}}\}\), to the nodes in \(V''_1 = \{v_{\gamma_1}^{(k+1)}(or \ v_{\gamma_2}^{(k+1)}), v_{\gamma_3}^{(k+1)}(or \ v_{\gamma_2}^{(k+1)}), \ldots, v_{\gamma_{k+1-m}}^{(k+1)}(or \ v_{\gamma_{k+1-m}}^{(k+1)})\}\), and to nodes in \(V'_2\).

Note that if necessary, we can use the nodes in \(V_2 - V'_2\) to form the \((k + 1 - m)\) paths. The remaining one node in \(V'_2\) can be reached from \(s\) through a \(d_{k+1} - (or \ \overline{d})\)-dimensional edge.

**Case 2:** node \(s\) in the subgraph \(Q_k\), i.e. \(s \in V_2\).

By Lemma 4.2, there are at most \(k\) distinct nodes in \(Q_k\) other than node \(s\) such that the paths from \(s\) to these \(k\) nodes are disjoint. Assume that \(l\) of \((k + 2)\) nodes are in \(Q_k\).

Let the other two nodes in \(HC_k\) be \(v_{a_1}\) and \(v_{a_2}\). The two disjoint paths from node 
\(s\) to nodes \(v_{a_1}\) and \(v_{a_2}\) are \((s, s^{(k+1)}(or \ s^{(k+1)}), \ldots, v_{a_1})\) and \((s, s^{(k+1)}(or \ s^{(k+1)}), \ldots, v_{a_2})\), respectively.

If \(k < l \leq k + 2\), we have \((l - k)\) nodes \(v_{a_i}\) in \(Q_k\) for \(1 \leq i \leq l - k\) that cannot be reached from node \(s\) by a disjoint path, and \((k + 2 - l)\) nodes \(v_{b_i}\) in \(HC_k\), for \(1 \leq i \leq k + 2 - l\), need to find disjoint paths from \(s\). With help of the \(d_{k+1}\) - (or \(\overline{d}\))-edges, the paths to these nodes from \(s\) are \((s, s^{(k+1)}(or \ s^{(k+1)}), \ldots, v_{a_1}^{(k+1)}(or \ v_{a_1}^{(k+1)}), v_{a_i})\) and \((s, s^{(k+1)}(or \ s^{(k+1)}), \ldots, v_{b_1})\), respectively.

When \(l = m\) for \(0 \leq m < k\), we follow the approach similar to Case 1 of this lemma. Let \(V'_1 = \{v_{a_1}, v_{a_2}, \ldots, v_{a(k+2)-m}\} \subseteq V_1\) and \(V'_2 = \{v_{b_1}, v_{b_2}, \ldots, v_{b_m}\} \subseteq V_2\).

Obviously, there are \(m\) disjoint paths from \(s\) to the node \(v_{b_i} \in V'_2\). For the nodes in 
\(V'_1\), we route node \(s\) to \((k - m)\) nodes in \(V''_2 = \{v_{\gamma_1}, v_{\gamma_2}, \ldots, v_{\gamma_{k-m}}\} \subseteq (V_2 - V'_2 - \{s\})\),

to the nodes in \(V''_2 = \{v_{\gamma_1}^{(k+1)}(or \ v_{\gamma_1}^{(k+1)}), v_{\gamma_2}^{(k+1)}(or \ v_{\gamma_2}^{(k+1)}), \ldots, v_{\gamma_{k-m}}^{(k+1)}(or \ v_{\gamma_{k-m}}^{(k+1)})\}\), then 
to nodes in \(V'_2\). Therefore, we have \((k - m)\) disjoint paths from \(s\) to \((k - m)\) nodes in 
\(V'_1\) with help of nodes in \(V''_2\), and \(V_1 - V'_1\) if necessary. Two paths from \(s\) to the
two remaining nodes in $V'_1$ can be obtained through nodes $s^{[k+1]}$ and $s^{(k+1)}$. □

**Example 4.6:** In $HC_3$, let $s = v_1$. Then there are four disjoint paths from $s$ to nodes $v_2$, $v_4$, $v_6$, and $v_7$, say $(v_1,v_2)$, $(v_1,v_3,v_4)$, $(v_1,v_5,v_6)$, and $(v_1,v_7,v_7)$. In $HC_4$, let $s = v_13$, and the five distinct nodes are $v_1$, $v_3$, $v_5$, $v_6$, and $v_11$. The five disjoint paths from $s$ to them are $(v_13,v_9,v_1)$, $(v_13,v_2,v_3)$, $(v_13,v_5)$, $(v_13,v_12,v_14,v_6)$, and $(v_13,v_15,v_11)$.

Next consider the Hamming cube $HC(N)$ of order $N$, where $2^{k-1} < N < 2^k$ and $k > 1$. There are $k + 1$ types of dimensional edges, $D = \{d_1, d_2, \ldots, d_k, \overline{d}\}$, in $HC(N)$. By the decomposition in Section 4.3, $HC(N) = \{HC_{k-1}, GIQ(N - 2^{k-1})\}$. Let $HC_{k-1} = (V_1, E_1)$ and generalized incomplete hypercube $GIQ(N - 2^{k-1}) = (V_2, E_2)$, where $V_1 = \{v_i | 0 \leq i \leq 2^{k-1} - 1\}$, $V_2 = \{v_i | 2^{k-1} \leq i \leq N - 1\}$. By Properties 4.3 and 4.4, we know that $DEG(v_i; N) \geq k$ for a node $v_i \in V_1$, and $2 \leq DEG(v_j; N) \leq k$ for $v_j \in V_2$. Therefore, the nodes in $V_2$ are candidates with the minimum degree of $HC(N)$, and $2 \leq \phi_{\text{min}}(HC(N)) \leq k$ for $k > 1$.

**Lemma 4.4.** Let $v_i \in V_2$ be a node of $GIQ(N - 2^{k-1})$, and $DEG(v_i; N) = m \geq \phi_{\text{min}}(HC(N))$. Then there are $m$ node-disjoint paths between $v_i$ and any node in $V_1$.

**Proof:** There are $(k + 1)$ types of dimensional edges, $D = \{d_1, d_2, \ldots, d_k, \overline{d}\}$ in network $HC(N)$. Since $DEG(v_i; N) = m$, node $v_i \in V_2$ has $m$ neighbors. Let nodes $v_1^{[k]}$ and $v_2^{[k]}$ in $V_1$ be the two neighbors of $v_{a_1}$ and $v_{a_2}$ in $V_2$ connected through the $d_k$- and $\overline{d}$-dimensional edges respectively, and the other $(m - 2)$ neighbors are $\{v_{a_3}, v_{a_4}, \ldots, v_{a_m}\}$ in $V_2$ connected through the $D - \{d_k, \overline{d}\}$ dimensional edges. The $(m - 2)$ neighbors are in turn connected to the corresponding nodes $v_3^{[k]}, v_4^{[k]}, \ldots, v_m^{[k]}$ in $V_1$ through $d_k$-dimensional edges. Let $V'_1 = \{v_1^{d_1}, v_2^{d_2}, v_3^{d_3}, \ldots, v_m^{d_m}\}$. Assume that node $v_{a} \in V_1 - V'_1$. Since the nodes in $V'_1$ are distinct and $m < k$, there exist $m$ disjoint paths from $v_{a}$ to the nodes in $V'_1$ by Lemma 4.3. Thus, the $m$ disjoint paths from $v_{a}$ to $v_{a}$ consist of two disjoint paths $(v_1, v_3^{[k]}, \ldots, v_{a})$ and $(v_1, v_3^{(d)}, \ldots, v_{a})$; and
(m - 2) disjoint paths from \(v_i\) to \(v_{\alpha_j} \in \{v_{\alpha_1}, v_{\alpha_2}, \ldots, v_{\alpha_m}\}\), to \(v_{\alpha_j} \in \{v_{\alpha_1}, v_{\alpha_2}, \ldots, v_{\alpha_m}\}\), and to \(v_\beta\). If node \(v_\beta \in \{v_{\alpha_1}, v_{\alpha_2}\}\) or \(\{v_{\alpha_1}, v_{\alpha_2}, \ldots, v_{\alpha_m}\}\), one of the \(m\) disjoint paths will become trivial - either the path \((v_i, v_\beta)\) or \((v_i, v_{\alpha_j}, v_\beta)\).

Lemma 4.5. Let nodes \(v_i\) and \(v_j\) be two arbitrary nodes in \(V_1 = \{v_i \mid 0 \leq i \leq 2^{k-1} - 1\}\) of \(HC(N)\). Then there are at least \(k\) disjoint paths between them.

Proof: The induced subgraph \(HC_{k-1} = (V_1, E_1)\) is a \((k - 1)\)-dimensional Hamming cube. By Lemma 4.1, there are at least \(k\) disjoint paths between any two nodes in \(HC_{k-1}\). □

Lemma 4.6. Let nodes \(v_i\) and \(v_j\) be two nodes in \(V_2 = \{v_i \mid 2^{k-1} \leq i \leq N - 1\}\) of Hamming cube \(HC(N)\), and \(\phi_{\text{min}}(HC(N)) \leq \deg(v_i; N) = \lambda \leq \mu = \deg(v_j; N)\). Then there exist \(\lambda\) disjoint paths between \(v_i\) and \(v_j\).

Proof: Let the \((k+1)\) types of dimensional edges in \(HC(N)\) be \(D = \{d_1, d_2, \ldots, d_k, \bar{d}\}\).

Assume that the Hamming distance between nodes \(v_i\) and \(v_j\) is \(\rho(v_i, v_j) = l\), where \(1 \leq l \leq k\), which determines the \(l\) dimensional edges from \(D\), say \(D^l\). Node \(v_i\) uses \(\lambda\) of the \((k + 1)\) dimensional edges to connect its neighbors, denoted as \(D^l_i\), whereas \(v_j\) uses \(\mu\) of such edges to connect its neighbors, denoted as \(D^l_j\).

We split the edge-set \(D^l_i\) into two subsets \(D^l_i\) and \(D^l_j\) according to the dimensional edges in \(D^l\) such that \(D^l_i \subseteq D^l\) and \(D^l_j \subseteq D^l\). Also, \(D^\lambda_j\) is divided into two subsets, say \(D^\lambda_i\) and \(D^\lambda_j\), in the same way. Note that \(D^\lambda_i \subseteq D^\lambda\). Let \(\lambda' \leq l\) be the number of elements in \(D^\lambda_i\). Then there are \(\lambda'\) disjoint paths of the form \((v_i, v'_1, \ldots, v'_t, v_j)\) between \(v_i\) and \(v_j\) by means of the edges in \(D^\lambda_i \cup D^\lambda_j\), where \(e = (v_i, v'_t) \in D^\lambda_i\) and \(e' = (v'_t, v_j) \in D^\lambda_j\). The other \((\lambda - \lambda')\) disjoint paths are of the form \((v_i, v'_t, \ldots, v'_t, v_j)\), where \(e = (v_i, v'_t) \in D^\lambda_i\) and \(e' = (v'_t, v_j) \in D^\lambda_j\) are the same dimensional edges for \(1 \leq t \leq (\lambda - \lambda')\). Therefore, there exist \(\lambda\) disjoint paths between \(v_i\) and \(v_j\). □

Example 4.7: Consider nodes \(v_{16}\) and \(v_3\) in the network \(HC(22)\) such that \(\phi_{\text{min}}(HC(22))\)
The five disjoint paths between them are given as $path_1 = (v_{16}, v_0, v_3)$, $path_2 = (v_{16}, v_{16}, v_{11}, v_3)$, $path_3 = (v_{16}, v_{17}, v_1, v_3)$, $path_4 = (v_{16}, v_{18}, v_2, v_3)$, and $path_5 = (v_{16}, v_{20}, v_4, v_3)$. Let $v_{21}$ and $v_{19}$ be another pair of nodes in $HC(22)$. There are four disjoint paths between them such that $path_1 = (v_{21}, v_{17}, v_{19})$, $path_2 = (v_{21}, v_{20}, v_{16}, v_{18}, v_{19})$, $path_3 = (v_{21}, v_{10}, v_8, v_{12}, v_{19})$, and $path_4 = (v_{21}, v_5, v_1, v_3, v_{19})$.

By Lemmas 4.4, 4.5, and 4.6, we have the following theorem.

**Theorem 4.1.** The minimum node-degree in network $HC(N)$ is given by

$$\phi_{\min}(HC(N)) \leq \kappa(HC(N)),$$

where $2^{k-1} < N < 2^k$ and $k > 1$.

As a result of Lemma 4.1 and Theorem 4.1, we conclude

**Corollary 4.2.** The Hamming cube $HC(N)$, for $N \geq 2$, is an optimally fault-tolerant network.

### 4.6.2. Node-Disjoint Paths

We have shown that Hamming cubes are optimally fault-tolerant networks since the vertex-connectivity is equal to the minimum degree. Consider the decomposition $HC_n = \{HC_2, Q_2, Q_3, \ldots, Q_{n-1}\}$. By Property 4.7, the nodes in $HC_i$ (or $Q_{i-1}$ if $i \geq 3$) have higher degrees than those in $HC_j$ (or $Q_{j-1}$), where $2 \leq i < j \leq n$. This implies that the number of node-disjoint paths between two nodes in $HC_i$ may be larger than that of the node-disjoint paths between two nodes, at least one of which is in $HC_j$.

Obviously, the number of paths between two distinct nodes $v_i$ and $v_j$ in a Hamming cube can be very large when common links are allowed to use. Thus, we simplify our analysis by restricting the reachable paths to only node-disjoint paths. Under this assumption, the number of node-disjoint paths between $v_i$ and $v_j$ is bounded by the minimum degree of the network. Let us find the number of node-disjoint paths be-
between two nodes in $HC_n$.

**Theorem 4.2.** In the network $HC_n$, let $HC_i$ and $HC_j$ be two subnetworks, where $2 \leq i \leq j \leq n$. Also let $v_a \in HC_i$ and $v_b \in HC_j$ such that the Hamming distance $\rho(v_a, v_b) = h$ and $1 \leq h \leq j$.

**Case 1:** For $0 < h < \lfloor \frac{j}{2} \rfloor$, there exist

1. $h$ paths of length $h$ and $(j + 1 - h)$ paths of length $h + 2$.
2. When $j < n$, there are $2(n - j)$ additional paths of length $h + 2$.

**Case 2:** For $\frac{j}{2} \leq h \leq j$, there exist

1. $h$ paths of length $h$ and $(j + 1 - h)$ paths of length $j + 1 - h$.
2. When $j < n$, the additional paths are as follows:
   - (2.2.1) If $n \leq 2h$, then two paths each of length $(j + 1 - h + i)$, where $0 < i \leq n - j$.
   - (2.2.2) If $n > 2h$, then two paths of length $(j + 1 - h + i)$ where $0 < i \leq 2h - j$, and $2(n - 2h)$ paths of length $h + 2$.

**Proof:** Let $P = \{p_i \mid 1 \leq i \leq h \text{ and } p_{i+1} < p_i\}$ be the set of bit positions in which node-labels of $v_a$ and $v_b$ differ. Note that each bit position $p_i \in P$ corresponds to a sheaf of $p_i$-dimensional $E1$-edges. So, let $P'$ be the set of all dimensional edges determined by the bit positions in $P$. In Case 1, $0 < h < \lfloor \frac{j}{2} \rfloor$, the $h$ paths of length $h$ are obtained by cyclically shifting the $p_i$-dimensional edges in $P'$ as follows.

1st path : $p_1, p_2, \ldots, p_{h-1}, p_h$

2nd path : $p_2, p_3, \ldots, p_h, p_1$

\vdots

ith path : $p_i, p_{i+1}, \ldots, p_{i-2}, p_{i-1}$

\vdots
Let $D = \{i, n_{i+1} \mid 0 \leq i \leq n-1\}$ be the set consisting of all dimensional edges in $HC_n$. Consider two subsets of $D$ as $D_1 = \{i \mid 0 \leq i \leq j - 1\} \cup \{n_j\}$ and $P_1 = \{d_i \mid 0 < i \leq j + 1 - h\} = D_1 - P'$. Then the $(j + 1 - h)$ paths of length $h + 2$ are obtained as $(d_1, p_1, p_2, \ldots, p_h, d_1)$, $(d_2, p_1, p_2, \ldots, p_h, d_2, \ldots)$, $\ldots$, $(d_{j+1-h}, p_1, p_2, \ldots, p_h, d_{j+1-h})$. Let $P_2 = \{i, n_{i+1} \mid j \leq i \leq n - 1\} \subset D$, and $d' \in P_2$. The $2(n - j)$ additional paths of length $h + 2$ have the form $(d', p_1, p_2, \ldots, p_h, d')$.

In Case 2.1, the $h$ paths of length $h$ can be obtained by the same analysis as in Case 1.1, while the $(j + 1 - h)$ paths of length $(j + 1 - h)$ are obtained by cyclically shifting the dimensional edges $(d_1, d_2, \ldots, d_{j+1-h})$ of $P_1$. Now consider Case 2.2.2. For each length $(j + 1 - h + i)$, there is a pair of disjoint paths given by

(i) $\underbrace{j, n_{j+1}}$ and $\underbrace{n_{j+1}, j}$ for $j + 1 - h + i = 2$,

(ii) $\underbrace{(j + i - 1), (j + i - 2), \ldots, j, n_{j+1}}$ and $\underbrace{n_{j+1}, (j + i - 2), \ldots, j, (j + i - 1)}$

for $2 < j + 1 - h + i < h + 2$.

Note that the second path is obtained from the first path by simply interchanging the first and the last dimensional edges. Let $P_3 = \{i, n_{i+1} \mid 2h \leq i < n\} \subset D$, and $d'' \in P_3$. The $2(n - 2h)$ paths of length $h + 2$ have the form $d'', p_1, p_2, \ldots, p_h, d''$. Thus Case 2.2.1 is a special case of Case 2.2.2.

**Example 4.8:** In the network $HC_5$, consider the following cases: (1) $v_\alpha = v_1$ and $v_\beta = v_3$, both in $HC_2$. (2) $v_\alpha = v_2 \in HC_2$ and $v_\beta = v_4 \in HC_3$. (3) $v_\alpha = v_0 \in HC_2$ and $v_\beta = v_7 \in HC_3$. Figures 4.4(a)–(c) show the node-disjoint paths between the corresponding nodes. In these figures, the label on each edge indicates the dimension of that edge, and the additional paths are distinguished by broken lines. By Theorem 4.2, the number of edges-disjoint paths between each pair of $v_\alpha$ and $v_\beta$ is listed in Table 4.4.
Figure 4.4: In network $HC_3$, the node-disjoint paths between (a) $v_1$ and $v_3$, (b) $v_2$ and $v_4$, and (c) $v_0$ and $v_7$.

An $f$-fault diameter of a network $G$ is defined as the worst-case diameter of $G$ after removing $f$ nodes from it. By Theorem 4.2 and the fact that Hamming cubes are optimally fault-tolerant, the following corollary is obtained.

**Corollary 4.3.** The $n$-fault diameter of the $n$-dimensional Hamming cube $HC_n$ is $\lceil \frac{n}{2} \rceil + 1$, which is equal to the diameter plus one.

A family of graphs $G$ is said to be *strongly resilient* [50] if the fault-diameter is at most $\text{diam}(G) + c$, where $c$ is a fixed constant independent of the order of $G$. Corollary 4.3 implies that the $n$-dimensional Hamming cube $HC_n$ is a strongly resilient network.
Table 4.4: The number of disjoint paths for Example 4.8.

<table>
<thead>
<tr>
<th>$v_a$</th>
<th>$v_b$</th>
<th>$h$</th>
<th>$j$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v_1$</td>
<td>$v_3$</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>6</td>
<td>-</td>
<td>9</td>
</tr>
<tr>
<td>$v_2$</td>
<td>$v_4$</td>
<td>2</td>
<td>3</td>
<td>-</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>$v_0$</td>
<td>$v_7$</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>5</td>
<td>-</td>
<td>8</td>
</tr>
</tbody>
</table>

with the constant $c = 1$.

4.6.3. Two-terminal Reliability

The two-terminal reliability [21], also called path reliability, between two distinct nodes is the probability of finding a reachable path between them.

Assume that faults occurring on edges of the path are independent. Let $p$ be the probability of an edge functioning normally. Let $E(A, B)$ denote the event that at least one of $A$ paths with the same length $B$ is operational. The probability of event $E(A, B)$ is $Pr[E(A, B)] = 1 - (1 - p^B)^A$. By Theorem 4.2, the two-terminal reliability $R$ between the nodes $v_a$ and $v_b$ in subgraphs $HC_i$ and $HC_j$, respectively, where $2 < i < j < n$, can be derived as follows.

Theorem 4.3.

Case 1: $j = n$

(i) For $0 < h < \left\lfloor \frac{n}{2} \right\rfloor$,

$$R = Pr[E(h, h)] + Pr[E(n + 1 - h, h + 2)] - Pr[E(h, h)] \cdot Pr[E(n + 1 - h, h + 2)].$$

(ii) For $\left\lfloor \frac{n}{2} \right\rfloor \leq h \leq n$,

$$R = Pr[E(h, h)] + Pr[E(n + 1 - h, n + 1 - h)] - Pr[E(h, h)] \cdot Pr[E(n + 1 - h, n + 1 - h)].$$
Case 2: \( j < n \)

(i) For \( 0 < h < \left\lfloor \frac{j}{2} \right\rfloor \),

\[
R = Pr[E(h, h)] + Pr[E(2n - j + 1 - h, h + 2)] - Pr[E(h, h)] \cdot Pr[E(2n - j + 1 - h, h + 2)].
\]

(ii) For \( \left\lfloor \frac{j}{2} \right\rfloor \leq h \leq n \),

If \( n \leq 2h \), \( R = Pr\{E(h, h) + E(j + 1 - h, j + 1 - h) + \sum_{i=1}^{n-h} E(2, j + 1 - h + i)\} \).

If \( n > 2h \), \( R = Pr\{E(h, h) + E(j + 1 - h, j + 1 - h) + \sum_{i=1}^{2h-j} E(2, j + 1 - h + i) + E(2(n - 2h), h + 2)\} \).

In Case 2, the two-terminal reliability \( R \) includes the probability of \( M \) multiple events. The general formula follows the principle of inclusion and exclusion \([49]\) and is given below.

\[
R = Pr[\sum_{1 \leq j \leq M} E_j] - Pr[\sum_{1 \leq j \leq k \leq M} E_j \cup E_k] + Pr[\sum_{1 \leq i < j \leq M} E_i \cap E_j \cap E_k] - \ldots
\]

\[+(-1)^M p\{E_1 \cap \ldots \cap E_M\} \]

Consider the worst case that node \( v_\beta \) is in subgraph \( Q_{n-1} \) (or \( HC_n \)). According to Theorem 4.2, there are \( m = n + 1 \) node-disjoint paths between \( v_\alpha \) and \( v_\beta \). The two-terminal reliability satisfies Case 1 of Theorem 4.3, which is the two-terminal reliability of the folded hypercubes \([36]\). In the \( n \)-dimensional hypercube \( Q_n \), there are \( h \) paths of length \( h \) and \( n - h \) paths of length \( h + 2 \) between any two nodes at a Hamming distance \( h \). The two-terminal reliability of \( Q_n \) is given by \( R_{Q_n} = Pr[E(h, h)] + Pr[E(n - h, h + 2)] - Pr[E(h, h)] \cdot Pr[E(n - h, h + 2)] \). Figures 4.5 and 4.6 show the two-terminal reliabilities of Hamming cubes and hypercubes, \( R_{HC_n} \) (the worst case) and \( R_{Q_n} \) with probabilities \( p = 0.8 \) and \( 0.6 \), respectively. Due to the additional \( E2 \)-edges, the curve of \( R_{HC_n} \) tends to concave upwards at values greater
than \( \frac{n}{2} \) when \( p \) decreases.

4.6.4. Container Quality

A container [55] is an abstract box which zooms in the capability of fault-tolerance between two distinct nodes in a network. The width and length of a container are respectively the number of node-disjoint paths between the nodes, and the length of the longest path in the box. Evidently, a wide but short container is preferred which means the network has high fault-tolerance and more efficient communication. By Theorem 4.2 and Corollary 4.3, the containers of two nodes \( v_\alpha \in HC_i \) and \( v_\beta \in HC_j \) is a \( (2n - j + 1) \times (h + 2) \) box, where \( 2 \leq i \leq j < n \) and \( \rho(v_\alpha, v_\beta) = h \). When \( j = h = n \), the container of the \( n \)-dimensional Hamming cube \( HC_n \) is a \( (n + 1) \times (\lceil \frac{n}{2} \rceil + 1) \) box. Furthermore, the container of \( HC_n \) will not increase.
its length under at least \( n \) faulty nodes.

The container quality \( [36] \) between two nodes at distance \( h \) is defined as \( \text{CON}(h) = \frac{A}{B} \), where \( A \) is the number of node-disjoint paths between the nodes and \( B \) is the average length of all node-disjoint paths between them. A higher value of \( \text{CON}(h) \) is favorable which implies that the container with a small "average" length has less communication delay. With the help of Theorem 4.2, the container quality \( \text{CON}(h) \) of the network \( HC_n \) can be easily derived.

Theorem 4.4.

Case 1: \( j = n \)

(i) For \( 0 < h < \left\lfloor \frac{n}{2} \right\rfloor \), \( \text{CON}(h) = \frac{(n+1)^2}{h(n-1)+2n+2}. \)

(ii) For \( \left\lfloor \frac{n}{2} \right\rfloor \leq h \leq n \), \( \text{CON}(h) = \frac{(n+1)^2}{2h^2-2(n+1)h+(n+1)^2}. \)
Case 2: \( j < n \)

(i) For \( 0 < h < \left\lfloor \frac{j}{2} \right\rfloor \), \( CON(h) = \frac{(2n-j+1)^2}{h(2n-j-1)-2j+4n+2} \).

(ii) For \( \left\lfloor \frac{j}{2} \right\rfloor \leq h \leq n \),

- If \( n \leq 2h \), \( CON(h) = \frac{(2n-j+1)^2}{h^2+(j+1-h)^2+(n-j)(n+j-2h+3)} \).
- If \( n > 2h \), \( CON(h) = \frac{(2n-j+1)^2}{h^2+(j+1-h)^2+(2h-j)(j+3)+2(n-2h)(h+2)} \).

When \( j = n \), \( CON(h) \) is given by the Case 1 which is also the container quality of folded hypercubes. As shown in [36], \( CON(h) \) for \( \left\lfloor \frac{n}{2} \right\rfloor \leq h \leq n \), is better than that of binary hypercubes because of the additional links.

4.6.5. Vulnerability

Akers and Krishnamurthy [4] proposed the notion of \textit{vulnerability} to classify the fault-tolerance of interconnection networks. Let \( N \) be the network order, \( F \) the number of faulty nodes, \( U \) the number of usable nodes which are the nodes in the largest connected component in the resultant network, and let \( I = N - F - U \) be the number of isolated nodes. The \textit{vulnerability} of a network is the maximum of \( \frac{I}{F} \), taken over all possible sets of faulty nodes. A network having a lower vulnerability is preferable. It has been shown that binary hypercube \( Q_n \) has \textit{poly-log} vulnerability, that lies between \( O(\log N) \) and \( O(\sqrt{\log N}) \), where \( N = 2^n \).

Since Hamming cube \( HC_n \) has node-connectivity \( n \), the lower bound of its vulnerability is \( \frac{1}{\log N+1} \) when all the \( n \) neighbors of a node are faulty. We know that in the decomposition \( HC_n = \{ HC_{n-1}, Q_{n-1} \} = \{ HC_{n-2}, Q_{n-2}, Q_{n-1} \} \), each node \( v_\alpha \) in the subgraph \( Q_{n-1} \) has two edges — one \((n-1)\)-dimensional \( E_1 \)-edge linked to node \( v_{\alpha \left\lfloor \alpha \right\rfloor n} \), and another \( n \)-dimensional \( E_2 \)-edge linked to node \( v_{\alpha (n)} \). Consider a subset of nodes \( S = \{ v_i \mid 2^{n-1} \leq i \leq 2^{n-1} + 2^{n-3} - 1 \} \). Let \( S_F \) be the set of faulty nodes consisting of all neighbors of the nodes in \( S \). Then \( F = 2^{n-1}, U = 2^{n-2}, and \)
\[ \begin{align*}
I &= N - F - U = 2^{n-2}, \text{ and the vulnerability of } HC_n \text{ is } \frac{I}{F} = \frac{1}{2}, \text{ a constant value. Any other choice of } S_F \text{ results in a value less than } \frac{1}{2}.
\end{align*} \]

4.7. Fault Diagnostic Algorithm

In this section, we design a testing algorithm for the injured \( n \)-dimensional Hamming cube \( HC_n \). A Hamming cube possessing some faulty components (nodes or links) is called an injured Hamming cube. The proposed algorithm can diagnose up to \( n + 1 \) faulty nodes (processors). Each healthy or non-faulty processor detects all of its neighbors. The testing table \( T \) is constructed to record the diagnostic results. A boolean variable \( t_{ij} \) denotes the outcome that processor \( v_i \) tests processor \( v_j \), where \( 0 \leq i, j \leq 2^n - 1 \), such that \( t_{ij} = 1 \) if \( v_i \) diagnoses \( v_j \) as faulty, otherwise \( t_{ij} = 0 \). An element \( T_{ij} \) located at the \( i \)th row and \( j \)th column of matrix \( T \) may contain the value of \( t_{ij} \) or \( u \), where \( u \) is an integer (say \( u = 2 \)), indicating \( T_{ij} \) is 'unknown' at the current time.

The diagnostic algorithm constructs the testing table row by row, starting at the 0th row\(^2\). By Property 4.8, let \( D = \{\alpha_0, \alpha_1, \ldots, \alpha_{\text{DEG}(u;2^n)-1}\} \) represent the set of links by which node \( v_i \) is connected to its neighbors. To determine elements \( T_{ij} \) for \( 0 \leq j \leq 2^n - 1 \), node \( v_i \) first diagnoses \( v_j \) which is its neighbor connected by the \( \alpha_0 \)-dimensional link. Assume that node \( v_j \) is found non-faulty. The two nodes then continue to test their respective neighbors in the \( \alpha_1 \) dimension. Next, all four members of the detected nodes (assuming they are all healthy) test their respective neighbors in the \( \alpha_2 \) dimension, and so on. This procedure continues until all members of detected nodes test their corresponding neighbors connected by \( \alpha_{\text{DEG}(u;2^n)-1} \).

\(^2\)Since the two-dimensional Hamming cube forms a complete graph \( K_4 \) of size four, we can choose any one of row \( i \) for \( 0 \leq i \leq 3 \), as the starting row.
In order to determine the element \( T_{ij} \), if nodes \( v_i \) and \( v_j \) are not neighbors, a healthy neighboring processor \( v_k \) of \( v_j \) (with \( T_{ik} = 0 \)) in a dimension \( \alpha_p \) help obtaining the value of \( T_{ij} \). The testing result is also written into \( T_{kj} \) to avoid repeated diagnoses. If \( v_k \) is found faulty (i.e. \( T_{ik} = 1 \)), then it cannot be used to determine \( T_{ij} \). The variable \( FN_j \) is flagged to notify this special case that \( v_j \) needs to be diagnosed by its other neighbor in the dimension \( \alpha_q \), where \( p > q \). The formal diagnostic algorithm for the injured \( HC_n \) is given as follows.

**Algorithm Diagnosis:** diagnose the faulty nodes in the injured \( HC_n \).

**Step 1** Initialize the testing table \( T \) and other variables.

- Set \( T_{ij} = u = 2 \) for \( 0 \leq i \neq j \leq 2^n - 1 \) and \( T_{ii} = 0 \) for \( 0 \leq i \leq 2^n - 1 \).
- Set \( FN_i = 0 \) for \( 0 \leq i \leq 2^n - 1 \). Set \( i = 0 \).

**Step 2** By Property 4.7, compute the dimensions of links that node \( v_i \) connects to its neighbors, \( D = \{ \alpha_0, \alpha_1, \ldots, \alpha_{\text{DEG}(v_i; 2^n)-1} \} \), and index \( l \) such that \( 2^{l+1} \leq i \leq 2^{l+2} - 1 \) for \( i \geq 4 \) and \( l = 0 \) for \( 1 \leq i \leq 3 \).

- Set variables \( d = l + 1 \), \( p = 0 \), and \( \text{branch} = 0 \).

**Step 3** For each node \( v_m \), \( 0 \leq m \leq 2^n - 1 \), check to see if \( T_{i,m} = 0 \), or 1.

- If \( T_{i,m} = 0 \) and \( T_{i,m_p} = u \), where node \( v_{m_p} \) is the neighbor of \( v_m \) in the dimension \( \alpha_p \), then set \( T_{i,m_p} = t_{m,m_p} \) and \( T_{m,m_p} = t_{m,m_p} \) and \( FN_{m_p} = 0 \).
- If \( T_{i,m} = 1 \) and \( T_{i,m_p} = u \), then set \( FN_{m_p} = u \).
- If \( \text{branch} = 2 \) or \( p = l \), then go to Step 5.

**Step 4** If \( p = \text{DEG}(v_i; 2^n - 1) \), then go to Step 6;

- else if \( p > l \), then \( \text{branch} = \text{branch} + 1 \) and \( p = p + 1 \);
- else \( p = p + 1 \). Repeat Step 3.

**Step 5** This step determines \( T_{ij} \) of blocked node \( v_j \) whose \( FN_j = u \).

- Let \( j = 0 \).
While \( j < 2^d \) do

If \( FN_j = u \), then search a neighbor \( v_k \) of node \( v_j \) in the dimension \( \alpha_q \)
for \( q < p \), such that \( T_{i,k} = 0 \). Let \( T_{i,j} = t_{k,j} \), \( T_{k,j} = t_{k,j} \), and \( FN_j = 0 \);
else \( j = j + 1 \).

If \( d < n \), then \( d = d + 1 \), \( p = p + 1 \), \( branch = 0 \), and go to Step 3.

**Step 6** If \( i = 2^n - 1 \), then stop; else \( i = i + 1 \), \( FN_j = 0 \) for \( 0 < j < 2^n - 1 \),
and go to Step 2.

Note that if a processor node \( v_i \) is faulty, then its testing results \( t_{ij} = y \in \{0,1,u\} \) are unreliable. Assuming that there are \( x \) number of faulty nodes in the injured Hamming cube \( HC_n \), the following lemma can be easily obtained.

**Lemma 4.7.** The testing table \( T \), constructed by Algorithm Diagnosis which diagnoses \( x \) faulty nodes in the injured \( HC_n \), satisfies:

1. the sum of elements \( T_{i,j} \), for \( 0 < j < 2^n - 1 \), in a row \( i \) is given by

\[
S_i = \begin{cases} 
  x & \text{if node } v_i \text{ is healthy;} \\
  y(2^n - 1) & \text{if node } v_i \text{ is faulty.}
\end{cases}
\]

2. the sum of elements \( T_{i,j} \), for \( 0 < i < 2^n - 1 \), in a column \( j \) is

\[
S_j = \begin{cases} 
  xy & \text{if node } v_j \text{ is healthy;} \\
  (x - 1)y + (2^n - x - 1) & \text{if node } v_j \text{ is faulty.}
\end{cases}
\]

Due to high connectivity of \( HC_n \), a specific node will be disconnected from others only when its \( n + 1 \) adjacent nodes (or links) become faulty. In this case, the isolated node cannot use its neighbors to construct the respective row and leave the elements in that row unknown (i.e., \( u = 2 \)). One can obtain the correct diagnosis by replacing the \( u \)'s by \( 0 \)'s and then using the property in Lemma 4.7.
Example 4.9: Let us illustrate how the diagnostic algorithm tests the faulty nodes in the Hamming cube $HC_3$. We consider two configurations of faults.

(1) Assume that the faulty modules occur in three nodes $v_1$, $v_3$, and $v_6$ marked by (X).

![Diagram of HC3 showing faulty nodes]

<table>
<thead>
<tr>
<th>$T_{ij}$</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

$0 \rightarrow 1$ ($T_{01} = 1$)
$0 \rightarrow 2$ ($T_{02} = 0$)
$1 \rightarrow 3$ ($T_{13} = 1; FN_3 = 0$)
$0 \rightarrow 4$ ($T_{04} = 1$)
$1 \rightarrow 5$ ($FN_5 = u$)
$2 \rightarrow 6$ ($T_{26} = 1; T_{56} = 1$)
$3 \rightarrow 7$ ($FN_7 = u$)
$0 \rightarrow 7$ ($T_{07} = 0$)
$2 \rightarrow 5$ ($T_{25} = 0, T_{55} = 0, FN_5 = 0$)

(2) Let nodes $v_6$, $v_3$, $v_5$, and $v_6$ become faulty. Healthy processors $v_4$ and $v_7$ are
surrounded by these faulty nodes. The resulting testing table is shown in Figure 4.7(f). Undetermined values of 2's are left in the respective rows and columns of \( v_4 \) and \( v_7 \).

**Theorem 4.5.** Algorithm Diagnosis can identify no more than \( n+1 \) faulty processors in the injured \( n \)-dimensional Hamming cube, \( HC_n \).

**Proof:** Assume that a starting node \( v_s \) is not faulty and it has the minimum degree \( n + 1 \). Let \( k \) be the number of faulty nodes in the injured \( HC_n \). First consider the case \( k \leq n \). There exists at least one non-faulty node \( v_t \) which is a neighbor of \( v_s \). Therefore, the algorithm will not be blocked. With the help of \( v_t \), the nodes detected to be fault-free proceed the testing of their neighbors. When the algorithm terminates, all nodes in \( HC_n \) have thus been determined appropriately. When \( k = n + 1 \), the algorithm is blocked only when all \( n+1 \) nodes (or links) adjacent to a specific node are disrupted by faults. In this case, the correct diagnosis can be achieved by replacing the unknown tests (u's) by the assumed healthy tests (0's) and performing the process according to Lemma 4.7. \( \square \)

**4.8. VLSI Layout Consideration**

The generation of Hamming cube networks based on the recursively defined adjacency of nodes suggests a building-block approach to the construction of the \( n \)-dimensional Hamming cube, \( HC_n \). By our decomposition, \( HC_n \) can be obtained by the meta-union of a \( HC_{\frac{n}{2}} \) and \( Q_{\frac{n}{2}} \) (Section 4.4). This leads to the following bisection width of \( HC_n \). The bisection width of a network is the minimum number of links whose removal disconnects the network into two equal-sized pieces.

**Theorem 4.6.** The bisection width of the \( n \)-dimensional Hamming cube \( HC_n \) is \( N \), where \( N = 2^n \).
Proof: Let \( HC_n = (V^1, E^1) \) and \( Q_n = (V^2, E^2) \). The nodes in sets \( V^1 \) and \( V^2 \) are \( V^1 = \{v_i \mid 0 \leq i \leq 2^5 - 1 \} \) and \( V^2 = \{v_i \mid 2^5 \leq i \leq 2^n - 1 \} \), respectively. By the meta-union operation, each node \( v_a \in V^2 \) is linked to two nodes \( v_{a[1]} \) and \( v_{a[n]} \) in \( V^1 \), where \( i_{[b]} = (b_k b_{k-1} \ldots b_j \ldots b_1) \) and \( i_{[m]} = (b_k b_{k-1} \ldots b_m \ldots b_1) \). Hence the theorem follows. \( \square \)

We make use of the following known result on the layout area of a network [77].

**Theorem 4.7.** Let \( S(N) \) be any monotonically nondecreasing function. A network of \( N \) nodes with a strong \( S(N) \) separator can be laid out in a square whose side is \( O(\max(\sqrt{N}, \Delta S(N))) \), where \( \Delta S(N) = \sum_{i=0}^{\log_2 N} 2^i S(N^{\frac{i}{4}}) \).

Since the cutset determined by the meta-union operation separates the \( n \)-dimensional Hamming cube \( HC_n \) into two subgraphs with equal number of nodes, it is a strong separator [77]. Let \( S(N) \) be a linear function of \( \frac{N}{4} \), then \( \Delta S(N) = 2N - 4(\sqrt{2})^{\log N} \). This yields the following result.

**Theorem 4.8.** An \( n \)-dimensional Hamming cube can be laid out in a square whose side is \( O(N) \) for \( N = 2^n \).

### 4.9. Summary

Hamming cubes are attractive hypercube-based networks with many desirable properties. These recursive networks have, unit incrementability and half of logarithmic diameter. They are optimally fault-tolerant and easily decomposable. For Hamming cubes, simple deterministic self-routing scheme can always find a shortest path of length no large than the diameter of the network.

It has been shown that Hamming cubes exhibit very good performance in terms of fault-tolerance and reliability measures. The \( n \)-dimensional Hamming cube \( HC_n \) is strongly resilient, whose \( n \)-fault diameter is its diameter plus one, i.e. \( [\frac{n}{2}] + 1 \). Since
Hamming cubes are recursively defined, there exist additional disjoint paths between two smaller labeled nodes. This implies the reliability and fault-tolerance of $HC_n$ are at least as good as that of folded hypercube of the same size, and better than that of the $n$-dimensional binary hypercube. Furthermore, our diagnostic algorithm can identify up to $n + 1$ faulty processors. Finally, $HC_n$ has constant vulnerability and can be laid out in an $O(N \times N)$ square, where $N = 2^n$. 
CHAPTER 5

ENHANCED GENERALIZED INCOMPLETE HYPERCUBES

In this chapter, we design another new family of networks, called enhanced generalized incomplete hypercubes. This family also belongs to the class of incremental Hamming graphs with a variant generator set. This topology is similar to the generalized incomplete hypercubes but with the additional links generated by the $\Omega_{n+1}$ set.

5.1. Network Definition

An enhanced generalized incomplete hypercubes of order $N$, denoted as $EGIQ(N) = (V, E)$, where $N = 2^n + m$ and $0 \leq m < 2^n$, is an undirected, connected graph in which $V = \{v_0, v_1, \ldots, v_{N-1} | v_i = BR(i)\}$ is the set of labeled nodes. The set $V$ is partitioned into several subsets $V_1 = V_1^1 \cup V_1^2$ and $V_2$ such that

1. $V_1 = \{v_\alpha | 0 \leq \alpha \leq 2^n - 1\}$ and $V_2 = \{v_\beta | 2^n \leq \beta \leq N - 1\}$,
2. $V_1^2 = \{v_\beta | v_\beta \in V_2\}$, and
3. $V_1^1 = V_1 - V_1^2$.

There exists an edge $e = (v_i, v_j) \in E$, if and only if any one of the following conditions is satisfied:

[C1]: $\rho(v_i, v_j) = 1$ for $v_i$ and $v_j$ in $V$.
[C2]: $v_i \in V_2$ and $v_j \in V_1^2$.
[C3]: $\rho(v_i, v_j) = n$ for $v_i$ and $v_j$ in $V_1^1$. 

68
Figure 5.1: Enhanced generalized incomplete hypercubes, $EGIQ(N)$ for $2 \leq N \leq 8$ and $N = 11$.

We will designate the edges defined by Conditions [C1], [C2], and [C3] as C1-links, C2-links, and C3-links, respectively. Clearly, the C1-links define the underlying (generalized incomplete) hypercubes topology of $EGIQ(N)$, while the C2- and C3-links define the additional enhanced links to generalized incomplete hypercubes $GIQ(N)$. Note that an existing C2-link has Hamming distance of $n + 1 = \lceil \log N \rceil$. Such an edge has a dimension of $n + 1$, while an enhanced C3-link is of $n$-dimensional since its Hamming distance is $n$. Figures 5.1(a)–(h) depict networks $EGIQ(N)$, for $2 \leq N \leq 8$ and $N = 11$. In this figure, the C2-links and C3-links are distinguished.
from C1-links by broken lines.

In $EGIQ(N)$, where $N = 2^n + m$ and $0 \leq m < 2^{n-1}$, consider the following cases depending on the value $m$.

**Case I: $m = 0$**

According to the definition, if $m = 0$ then all additional enhanced links are C3-links.

**Case II: $0 < m < 2^{n-1}$**

When $0 < m < 2^{n-1}$, let $e = (v_i, v_j)$ be a C3-link in $EGIQ(N)$, where $\rho(v_i, v_j) = n$ and $j = i^{(n)}$. Assume that $v_k$ is a newly added node, and it is connected to $v_j$ through a C2-link $e' = (v_k, v_j)$. Clearly, $j = k$. In the new network $EGIQ(N + 1)$, the C3-link $e$ is replaced by the C2-link $e'$. This is called a link replacement. For example, as shown in Figures 5.1(c) and (d), link $e = (v_0, v_3)$ is replaced by $e' = (v_4, v_3)$ in which $v_4$ is the newly added node to $EGIQ(4)$. Interestingly, the incidence of a C2-link and a C3-link to a node are exhibited exclusively. A node can have either one incident C2- or C3-link to its neighbor, or neither, but not both. For necessity, the incident nodes $v \in A = V_1 = \{v_\alpha | 0 \leq \alpha \leq 2^n - 1\}$ of additional enhanced links are identified by the following three subsets. Let $A_1 = \{v_i | 0 \leq i \leq m - 1\}$, $A_2 = \{v_i | 2^{n-1} - m \leq i \leq 2^{n-1} - 1\}$, and $A_3 = A - A_1 - A_2$. The nodes in $A_1$ have no enhanced edges while the nodes in $A_2$ and $A_3$ have C2- and C3-links, respectively.

**Case III: $2^{n-1} \leq m < 2^n$**

When $2^{n-1} \leq m < 2^n$, all enhanced links are C2-links and no link replacement occurs.

Considering the special case when $m = 0$, the network $EGIQ(N)$ consists of $2^n$
nodes, called the \textit{n-dimensional} enhanced generalized incomplete hypercube, denoted as \( EGIQ_n \). By the definition of enhanced links, the \( 2^{n-1} \) number of \( C3 \)-links in \( EGIQ_n \) exactly correspond to the same number of \textit{complementary} edges in the folded hypercube, \( FQ_n \) [36]. Thus, \( EGIQ_n \) is identical with \( FQ_n \). Furthermore, consider the case \( m = 2^k \) and \( 0 \leq k < n \). We can decompose the node-set into \( 2^{n-k} + 1 \) subsets, \( S_i \), for \( 0 \leq i \leq 2^{n-k} \). The replaced of \( C3 \)-links in a link-replacement are incident to nodes in the 0th subset \( S_0 \), and the replacement of \( C2 \)-links takes place between two corresponding nodes in the subsets \( S_{2^{n-k}+1} \) and \( S_{2^{n-k}} \), respectively. Therefore, network \( EGIQ(2^n + 2^k) \) is an enhanced incomplete hypercube \( EIQ_k \) [19]. Hence, we have the following theorem.

\textbf{Theorem 5.1.} The enhanced generalized incomplete hypercube \( EGIQ(N) \) of order \( N \) is an folded hypercube \( FQ_n \), when \( N = 2^n \) and \( n \geq 0 \), and is an enhanced incomplete hypercube \( EIQ_k \), when \( N = 2^n + 2^k \) and \( 0 \leq k < n \).

Many structural and communication properties of folded hypercubes and enhanced incomplete hypercubes have been explored in [19, 36, 76]. Thus, we are interested in enhanced generalized incomplete hypercubes \( EGIQ(N) \) when the network order \( N \) is not the form \( 2^n \) or \( 2^n + 2^k \).

\textbf{5.2. Topological Properties}

The following provides several properties of \( EGIQ \)'s which include link complexity, node degree, and diameter. As shown in Theorem 5.1, network \( EGIQ(N) \) for \( N = 2^n \) is the folded hypercubes. Therefore, its link complexity is \( E(N) = \frac{N}{2} \log N + \frac{N}{2} \), where \( N = 2^n \) and \( n \geq 1 \). Property 5.1 provides a general formula for computing the number of links in the \( EGIQ \)'s of an arbitrary order \( N \).

\textbf{Property 5.1.} Let \( N = 2^{p_n} + 2^{p_{n-1}} + \ldots + 2^{p_1} \) and \( p_i > p_{i-1} > \ldots > p_1 \).
Then the number of edges in $EGIQ(N)$ is $E(N) = \sum_{j=1}^{q} p_j 2^{p_j - 1} + \sum_{j=1}^{q-1} \sum_{i=1}^{j} 2^{p_i} + e$, where

$$e \begin{cases} \sum_{j=1}^{q-1} 2^{p_j} & \text{for } 2^{p_1} < N < 2^{p_1} + 2^{p_1-1}; \\ 2^{p_1} + 2^{p_1-1} & \text{for } 2^{p_1} + 2^{p_1-1} \leq N < 2^{p_1+1}. \end{cases}$$

**Proof:** The node-set of $EGIQ(N)$ can be partitioned into $q$ subsets, $V_q = \{v_0, v_1, \ldots, v_{2^{p_1}-1}\}$ and $V_l = \{v_j | \sum_{i=l+1}^{q} 2^{p_i} \leq j \leq (\sum_{i=l}^{q} 2^{p_i}) - 1\}$ for $1 \leq l \leq q-1$. The number of $C1$-links is given by $\sum_{j=1}^{q} p_j 2^{p_j - 1} + \sum_{j=1}^{q-1} \sum_{i=1}^{j} 2^{p_i}$. By the link-replacement, the total number of enhanced $C2$- and $C3$-links is $e = 2^{p_1-1}$ for $2^{p_1} < N < 2^{p_1} + 2^{p_1-1}$. When $2^{p_1} + 2^{p_1-1} \leq N < 2^{p_1+1}$, the enhanced links are $\sum_{j=1}^{q-1} 2^{p_j}$ number of $C2$-links. Hence, the property follows. □

**Example 5.1:** In networks $EGIQ(11)$, $N = 11 = (1011)_2$, so that $q = 3$, $p_3 = 3$, $p_2 = 1$, and $p_1 = 0$. Similarly, for $N = 14 = (1110)_2$, $p_3 = 3$, $p_2 = 2$, and $p_1 = 1$. Hence by Property 5.1, $E(11) = 21$ and $E(14) = 31$, respectively.

**Property 5.2.** Let $2^n < N < 2^{n+1}$ for $n > 1$ and let $N = \sum_{i=1}^{l} 2^{p_i}$, where $1 \leq l \leq n+1$ and $p_l > p_{l-1} > \ldots > p_1$. Then in $EGIQ(N)$, the degree of a node $v_\alpha$ is obtained as follows.

(1) for $2^n < N \leq 2^n + 2^{n-1}$:

$$DEG(v_\alpha; N) = \begin{cases} p_{l-1} + i + 2 & \text{for } \sum_{j=i+1}^{l} 2^{p_j} \leq \alpha < \sum_{j=i+1}^{l} 2^{p_j} + \sum_{j=1}^{l-i-1} 2^{p_j}; \\ p_{l-1} + i + 1 & \text{for } \sum_{j=i+1}^{l} 2^{p_j} + \sum_{j=1}^{l-i-1} 2^{p_j} \leq \alpha < \sum_{j=i}^{l} 2^{p_j}; \\ p_l + 1 & \text{for } \sum_{j=1}^{l} 2^{p_j} \leq \alpha < \sum_{j=1}^{l} 2^{p_j}; \\ p_l + 1 & \text{for } 0 \leq \alpha < 2^{p_l}; \end{cases}$$
(2) for $2^n + 2^{n-1} < N < 2^{n+1}$:

$$DEG(v; N) = \begin{cases} 
  p_{i-i} + i + 2 & \text{for } \sum_{j=i-i+1}^{i} 2^{p_j} \leq \alpha < \sum_{j=i-i+1}^{i} 2^{p_j} + \sum_{j=1}^{i-i-1} 2^{p_j}; \\
  p_{i-i} + i + 1 & \text{for } \sum_{j=i-i+1}^{i} 2^{p_j} + \sum_{j=1}^{i-i-1} 2^{p_j} \leq \alpha < \sum_{j=i-1}^{i} 2^{p_j}; \\
  p_{i+1} + 1 & \text{for } \sum_{j=2}^{i} 2^{p_j} \leq \alpha < \sum_{j=1}^{i} 2^{p_j}; \\
  DEG(v; 2^n) + 2 & \text{for } \mathcal{S}_2 \cap \mathcal{S}_3; \\
  DEG(v; 2^n) + 1 & \text{for } (\mathcal{S}_2 \cup \mathcal{S}_3) - (\mathcal{S}_2 \cap \mathcal{S}_3); \\
  DEG(v; 2^n) & \text{for } \mathcal{S}_1 - (\mathcal{S}_2 \cup \mathcal{S}_3), 
\end{cases}$$

where $1 \leq i \leq l - 2$, $\mathcal{S}_1 = \{\alpha \mid 0 \leq \alpha < 2^m\}$, $\mathcal{S}_2 = \{\alpha \mid 0 \leq \alpha < \sum_{j=1}^{i-1} 2^{p_j}\}$, and $\mathcal{S}_3 = \{\alpha \mid 2^{p_i - \sum_{j=1}^{i-1} 2^{p_j}} \leq \alpha < 2^{p_i}\}$.

Proof: By counting the incident edges of a node, the property can be obtained by a similar analysis as for Property 4.4. \(\square\)

**Example 5.2:** Consider networks $EGIQ(11)$ and $EGIQ(13)$. By Property 5.2,

$$DEG(v; 11) = \begin{cases} 
  4 & \text{for } \alpha = 8; \\
  3 & \text{for } \alpha = 9; \\
  3 & \text{for } \alpha = 10; \\
  4 & \text{for } \alpha \in \{0, 1, \ldots, 7\} 
\end{cases}$$

$$DEG(v; 13) = \begin{cases} 
  5 & \text{for } \alpha = 8; \\
  4 & \text{for } \alpha = 9, 10, 11; \\
  3 & \text{for } \alpha = 12; \\
  5 & \text{for } \alpha \in \{3, 4\}; \\
  4 & \text{for } \alpha \in \{0, 1, 2, 5, 6, 7\} 
\end{cases}$$

**Property 5.3.** The diameter of $EGIQ(N)$, where $2^n < N < 2^{n+1}$ and $n \geq 1$, is given by $\text{diam}(EGIQ(N)) \leq \lceil \frac{n}{2} \rceil + 1 = \lceil \frac{\log_2 N}{2} \rceil + 1$. 
5.3. Network Decompositions

Let \( N = \sum_{i=1}^{l} 2^{p_i} \), where \( 1 \leq l \leq n + 1 \) and \( p_{i+1} > p_i \). As shown in Section 4.3, the generalized incomplete hypercube \( GIQ(N) \) and Hamming cube \( HC(N) \) can be decomposed as \( GIQ(N) = \{Q_{p_l}, Q_{p_{l-1}}, \ldots, Q_{p_1}\} \), and \( HC(N) = \{HC_{p_l}, Q_{p_{l-1}}, \ldots, Q_{p_1}\} \), respectively. We can easily decompose network \( EGIQ(N) \) in a similar way.

Consider the following two cases: (1) \( 2^n < N \leq 2^n + 2^{n-1} \) and (2) \( 2^n + 2^{n-1} < N < 2^{n+1} \). For the first case, network \( EGIQ(N) \) has both C3-links and C2-links.

Therefore, the subgraph induced by the node-subset \( V^{p_l} \) forms a \( p_l \)-dimensional incomplete folded hypercube, \( IFQ_{p_l} \), (a spanning subgraph of the folded hypercube \( FQ_{p_l} \) with some C3-links absent), denoted as \( EGIQ(N) = \{IFQ_{p_l}, Q_{p_{l-1}}, \ldots, Q_{p_1}\} \).

For the second case, all enhanced links in \( EGIQ(N) \) are C2-links, so the node-subset \( V^{p_l} \) forms a \( p_l \)-dimensional hypercube. Thus, the decomposition of this case is the same as that of \( GIQ(N) \). Figure 5.2 shows the decomposition of \( EGIQ(11) \) and
5.4. Self-routing Schemes

In this section, we design a simple routing scheme (one-to-one communication) for enhanced generalized incomplete hypercubes $EGIQ(N)$, where $2^n < N < 2^{n+1}$ and $n \geq 1$. We first divide the node-set $V = \{v_i \mid 0 \leq i \leq N - 1\}$ into two subsets, $V_1 = \{v_i \mid 0 \leq i \leq 2^n - 1\}$ and $V_2 = \{v_i \mid 2^n \leq i \leq N - 1\}$. Then according to the membership of the source node $v_i$ and destination node $v_j$, we consider four different cases: (1) $v_i, v_j \in V_1$, (2) $v_i \in V_1$ and $v_j \in V_2$, (3) $v_j \in V_1$ and $v_i \in V_2$, and (4) $v_i, v_j \in V_2$.

Let $v_i$ and $v_j$ be two distinct nodes in the network $EGIQ(N)$ and $P = \{p_\alpha \mid 1 \leq \alpha \leq h\}$ be the set of bit positions in which labels of $v_i$ and $v_j$ differ. As discussed in Section 5.1, the nodes in set $A_1 = \{v_i \mid 0 \leq i \leq m - 1\}$ have no enhanced edges. While the nodes in sets $A_2$ and $A_3$ have $C2$- and $C3$-links, respectively, where $A_2 = \{v_i \mid 2^{n-1} - m \leq i \leq 2^n - 1\}$ and $A_3 = A - A_1 - A_2$.

The routing for Case (1) is given as follows.

**Routing Scheme RS_EGIQ1**: Route node $v_i$ to $v_j$, both in set $V_1$.

1. Compute the Hamming distance $\rho(v_i, v_j) = h$.
2. If $(h \leq \lfloor \frac{h}{2} \rfloor)$ then begin
   2.1. Compute the set $P$ for nodes $v_i$ and $v_j$.
   2.2. Let $x_\alpha$ be the non-negative integer complementing the $p_\alpha$th bit of integer $i$, i.e., $x_\alpha = i^{\lfloor p_\alpha \rfloor}$, where $p_\alpha \in P$ and $1 \leq \alpha \leq h$.
   2.3. Route node $v_i$ to $v_j$ along the path $(v_i, v_{x_1}, v_{x_2}, \ldots, v_{x_{h-1}}, v_j)$.
   end
3. else begin
3.1 \textbf{If} \((v_i \in A_1)\) \textbf{then begin} \{ node \(v_i\) has no enhanced links \}

3.1.1 Let \(v_t = v_i(n+1)\) and \(v_t' = v_i(n+1)\).

3.1.2 Compute the set \(P\) for nodes \(v_t\) and \(v_j\).

3.1.3 Compute \(x_\alpha\) as in Line 2.2.

3.1.4 Route node \(v_i\) to \(v_j\) along the path \((v_i, v_t, v_{x_1}, v_{x_2}, \ldots, v_{x_{n-1}}, v_j)\).

\textbf{end}

3.2 \textbf{If} \((v_i \in A_2)\) \textbf{then begin} \{ node \(v_i\) has a C2-link \}

3.2.1 Let \(v_t = v_i(n+1)\) and \(v_t' = v_i(n+1)\).

3.2.2 Repeat Lines 3.1.2—3.1.4.

\textbf{end}

3.3 \textbf{If} \((v_i \in A_3)\) \textbf{then begin} \{ node \(v_i\) has a C3-link \}

3.3.1 Let \(v_t = v_i[n]\).

3.3.2 Compute the set \(P\) for nodes \(v_t\) and \(v_j\).

3.3.3 Compute \(x_\alpha\) as in Line 2.2.

3.3.4 Route node \(v_i\) to \(v_j\) along the path \((v_i, v_t, v_{x_1}, v_{x_2}, \ldots, v_{x_{n-1}}, v_j)\).

\textbf{end}

\textbf{end}

With help of Routing Scheme RS.EGIQ1, we next consider the remaining cases. Note that a node \(v_i\) in set \(V_1\) has 0 in its most significant bit, that is \(BR_{n+1}(i) = 0\), while the node in set \(V_2\) has \(BR_{n+1}(i) = 1\).

**Routing Scheme RS.EGIQ2:** Route node \(v_i\) to \(v_j\) in network \(EGIQ(N)\), where \(2^n < N < 2^{n+1}\) and \(n \geq 1\).

1 \textbf{If} \((BR_{n+1}(i) = 0)\) \textbf{then begin}

1.1 \textbf{If} \((BR_{n+1}(j) = 0)\) \textbf{then}

Since nodes \(v_i\) and \(v_j\) are in \(V_1\), use Routing Scheme RS.EGIQ1.
else begin \( BR_{n+1}(j) = 1 \)

1.2.1 Compute the Hamming distance \( \rho(v_i, v_j) = h \).

1.2.2 If \( h \leq \left\lfloor \frac{n}{2} \right\rfloor \) then \( v_t = v_{j(n+1)} \) else \( v_t = v_{j(n+1)} \).

1.2.3 Compute the set \( P \) for nodes \( v_i \) and \( v_t \).

1.2.4 Compute \( x_\alpha \) as in Line 2.2 of Routing Scheme RS.EGIQ1.

1.2.5 Route node \( v_i \) to \( v_j \) along the path \((v_i, v_{x_1}, v_{x_2}, \ldots, v_{x_{n-1}}, v_t, v_j)\).

end

end

2 If \( BR_{n+1}(i) = 1 \) then begin

2.1 if \( BR_{n+1}(j) = 0 \) then use the reverse routing path of the case when \( BR_{n+1}(i) = 0 \) and \( BR_{n+1}(j) = 1 \) as in Lines 1.2.1–1.2.5.

else begin

2.2.1 Compute the Hamming distance \( \rho(v_i, v_j) = h \).

2.2.2 If \( h \leq \left\lfloor \frac{n}{2} \right\rfloor \) then Repeat Lines 2.1–2.3 of Routing Scheme RS.EGIQ1.

2.2.3 else begin

2.2.3.1 Let \( v_t = v_{i(n+1)} \) and \( v_{t'} = v_{j(n+1)} \).

2.2.3.2 Compute the set \( P \) for nodes \( v_t \) and \( v_{t'} \).

2.2.3.3 Compute \( x_\alpha \) as in Line 2.2.

2.2.3.4 Route node \( v_i \) to \( v_j \) along the path \((v_i, v_{t}, v_{x_1}, v_{x_2}, \ldots, v_{x_{n-1}}, v_{t'}, v_j)\).

end

end

end

Example 5.3: Consider to route the following three pairs of nodes, \((v_0, v_{15})\), \((v_6, v_{13})\), and \((v_{10}, v_{20})\), in EGIQ(22). The Hamming distances between the corresponding node-pairs are \( \rho(v_0, v_{15}) = 4 \), \( \rho(v_6, v_{13}) = 3 \), and \( \rho(v_{10}, v_{20}) = 4 \), respectively.
(1) Since \( v_0, v_{15} \in V_1 \) and \( v_0 \in A_1 \), the routing path from \( v_0 \) to \( v_{15} \) is \( (v_0, v_{16}, v_{15}) \) by Routing Scheme RS.EGIQ1.

(2) Also by Routing Scheme RS.EGIQ1 but with \( v_6 \in A_3 \), the routing path from \( v_6 \) to \( v_{13} \) is \( (v_6, v_9, v_{13}) \).

(3) Since \( v_{10} \in V_2 \) and \( v_{20} \in V_2 \), the routing path from \( v_{10} \) to \( v_{20} \) is \( (v_{10}, v_{11}, v_{20}) \) by Routing Scheme RS.EGIQ2.

Since the routing path between any two nodes is determined by their labeled addresses, the routing schemes for enhanced generalized incomplete hypercubes are self-routing.

5.5. Summary

In this chapter, we have designed another new family of networks, called the enhanced generalized incomplete hypercubes. It has been shown that the folded hypercube and enhanced incomplete hypercube are special cases of enhanced generalized incomplete hypercubes. Therefore, this family of networks can be viewed as "enhanced" generalized incomplete hypercubes [26, 72] with extra links or "generalized" folded hypercubes [36] with the incrementability of one. Enhanced generalized incomplete hypercubes have half of logarithmic diameter. A simple routing scheme is designed, whose routing paths are always bounded by the network diameter has been designed.
CHAPTER 6

COMMUNICATIONS IN GENERALIZED INCOMPLETE HYPERCUBES

In this chapter, we construct two types of broadcasting (one-to-all communication) trees, spanning trees (ST's) and multiple spanning trees (MUST's), for generalized incomplete hypercubes. Both the trees are directed and edge-disjoint, and can be rooted at arbitrary node. The broadcasting data is sent from the root (source node) of the trees to the other nodes (destination nodes) following the direction of edges. The broadcasting scheme by using such trees is evaluated by the one-port and all-port communication models.

The results explored in this chapter will be used for the discussion in the following chapters which construct broadcasting trees in Hamming cubes and enhanced generalized incomplete hypercubes.

6.1. Spanning Trees in GIQ(N)

Let us first introduce the embedded spanning trees in the binary hypercubes (Q_n), called binomial spanning trees (BST’s), due to Johnson and Ho [47].

6.1.1. Binomial Spanning Trees (BST’s)

A binomial tree is defined recursively as follows. A 0-level binomial tree has one node. An l-level binomial tree is constructed from two (l - 1)-level binomial trees by making either of their roots as the new root and adding a new edge between the new root and the other root. The BST rooted at node v_0 in Q_n is constructed by connecting a node v_i for 0 ≤ i ≤ n - 1, to its neighbors whose addresses are obtained
by complementing any leading 0 bit in the binary representation of $i$. Note that the position of a complemented bit also defines the dimension of the edge connecting node $v_i$ to its neighbor. Figure 6.1(a) shows the BST rooted at the node $v_0$ in $Q_4$. The label on an edge indicates the dimension of that edge.

A BST rooted at an arbitrary node $s$ (a source node) can be simply translated from the BST rooted at the node $v_0$. The address of a node $v_j$ in the BST rooted at node $s$ is obtained by a bitwise Exclusive-OR operation on the address of its corresponding node $v_i$ in the BST rooted at node $v_0$ with the address of node $s$. Precisely speaking, let $s = BR(s) = (a_n, a_{n-1}, \ldots, a_1)$ and $v_i = BR(i) = (b_n, b_{n-1}, \ldots, b_1)$. Then $v_j = BR(j) = (c_n, c_{n-1}, \ldots, c_1)$, where $c_m = a_m \oplus b_m$ for $1 \leq m \leq n$. Figure 6.1(b) shows the BST rooted at $v_{12}$ in $Q_4$. The first neighbor $v_{13}$ of the root $v_{12}$ is obtained from $v_1 \oplus v_{12}$, where $v_1$ is the first neighbor of the root $v_0$ in Figure 6.1(a). Note that due to the translation of bitwise Exclusive-OR operation, the dimensions of tree edges in the BST's do not change. Thus, one can regard the BST rooted at node $v_0$ as a template for the construction of the embedded binomial spanning trees.

Figure 6.1: The BST's rooted at nodes $v_0$ and $v_{12}$ respectively in $Q_4$. 
rooted at an arbitrary node in the binary hypercubes.

6.1.2. Spanning Trees $ST_{GIQ(N)}$

Now let us design an algorithm to construct the spanning trees $ST_{GIQ(N)}$ for generalized incomplete hypercubes, $GIQ(N)$, where $2^{k-1} < N < 2^k$ and $k > 1$. The roots (the source nodes) of $ST_{GIQ(N)}$ can be an arbitrary. Let $GIQ(N) = \{Q_{p_1}, Q_{p_1-1}, \ldots, Q_{p_1}\}$. The $ST_{GIQ(N)}$ is constructed depending on the membership of the source node in its induced subgraphs.

Algorithm $ST_{-GIQ}$: Construct the $ST_{GIQ(N)}$ of $GIQ(N)$, where $2^{k-1} < N < 2^k$.

begin
1 Let $GIQ(N) = \{Q_{p_1}, Q_{p_1-1}, \ldots, Q_{p_1}\}$, where $N = \sum_{i=1}^{l} 2^{p_i}$, $1 \leq l \leq k$ and $p_{i+1} > p_i$.
2 Assume that the source node $s_i$ is in $Q_{p_i}$.
3 Construct the BST of $Q_{p_i}$ rooted at $s_i$, denoted as $BST_{p_i}$.
4 Let $U_{p_i} = V_{p_1} \cup V_{p_2} \ldots \cup V_{p_{i-1}}$.
5 for each node $v_i \in U_{p_i}^{-1}$ do
   make $v_i$ the child of $v_{i[p_i]}$ in $BST_{p_i}$.
6 for $j = i + 1$ to $l$ do
   begin
   6.1 Construct the BST of $Q_{p_j}$ rooted at $s^{[p_j]}_i$.
   6.2 Append $BST_{p_j}$ to $BST_{p_i}$ by making $s^{[p_j]}_i$ the child of $s_i$ through the $p_j$-dimensional edge.
   end
end.

In Algorithm $ST_{-GIQ}$, we first locate the source node $s_i$ at its subgraph $Q_{p_i}$ and construct the BST of $Q_{p_i}$ rooted at $s_i$. The nodes in the subgraphs $\{Q_{p_i-1}, Q_{p_i-2}, \ldots, Q_{p_1}\}$
are appended them to the BST of \( Q_{p_i} \), through \( p_i \)-dimensional edges. We also determine the neighbors of \( s_i \), namely \( s_i^{[p_j]} \), in the subgraph \( Q_{p_j} \), where \( i + 1 \leq j \leq l \), and construct the BST's of each \( Q_{p_j} \), denoted as \( BST_{p_j} \), and rooted at the corresponding nodes \( s_i^{[p_j]} \). After unifying all \( BST_{p_j}'s \) with \( BST_{p_i} \), we have constructed the required spanning tree \( ST_{GIQ(N)} \) of \( GIQ(N) \).

![Figure 6.2: The \( ST_{GIQ(N)}(15) \) rooted at node \( v_{11} \).](image)

**Property 6.4.** The height of \( ST_{GIQ(N)} \) is at most \( \lceil \log N \rceil = k \).

**Proof:** Let us consider the following two cases depending on the membership of the source node \( s_i \) in the subgraphs \( \{ Q_{p_1}, Q_{p_{i-1}}, \ldots, Q_{p_i} \} \).

**Case 1:** \( s_i \in Q_{p_i} \).

By Algorithm ST_.GIQ, we first construct the BST\(_{p_i}\) of \( Q_{p_i} \) which is rooted at \( s_i \). Note that the height of BST\(_{p_i}\) is \( p_i = k - 1 \). Then append each node in the subgraphs \( \{ Q_{p_{i-1}}, Q_{p_{i-2}}, \ldots, Q_{p_1} \} \) to BST\(_{p_i}\). When the node \( s_i \) is added, the height of BST\(_{p_i}\) will be increased by one.

**Case 2:** \( s_i \notin Q_{p_i} \).

Assume that \( s_i \) is in \( Q_{p_i} \). By Algorithm ST_.GIQ, we construct BST\(_{p_i}\) and include the node in the subgraphs \( \{ Q_{p_{i-1}}, Q_{p_{i-2}}, \ldots, Q_{p_1} \} \). The height of BST\(_{p_i}\) is at most \( p_i + 1 \). Among the BST\(_{p_j}\) rooted at \( s_i^{[p_j]} \), where \( i + 1 \leq j \leq l \), the BST\(_{p_i}\) has the
maximum value of height, $p_l$. After appending these $BST_{p_i}$ to $BST_{p_l}$, the height of $BST_{p_l}$ becomes $\max\{p_i + 1, p_l + 1\} = p_l + 1 = k$.

By Cases 1 and 2, the property follows. □

Figure 6.2 shows the spanning tree $ST_{GIQ(15)}$ in $GIQ(15)$ rooted at node $v_{11}$. It can be easily seen that the maximum degree $\phi_{\text{max}}(ST_{GIQ(N)}) \leq k$. Consider two cases for the one-port communication in $ST_{GIQ(N)}$: the source node $s_i \in Q_{p_l}$ or $s_i \notin Q_{p_l}$. For the first case, element $i$ for $0 \leq i < M$ is sent to the children of $s_i$ through the edges along dimensions $0, 1, \ldots, p_l - 1, p_l$ (if it exists) during a time period $t_p = k$. For the second case, element $i$ for $0 \leq i < M$ is sent to the children through the edges along dimensions $p_l, p_l - 1, \ldots, p_{l+1}, 0, 1, \ldots, p_i - 1$ also during a time period $t_p = k$. Therefore, the total time for the one-port broadcast is thus $T = \lceil \log N \rceil \times M$. By Property 6.4, the height of $ST_{GIQ(N)}$ is $k$. The total broadcast time is $T = M + \lceil \log N \rceil - 1$ with all-port communication.

6.2. Multiple Spanning Trees in $GIQ(N)$

In this section, we will construct the embedded multiple spanning trees (MUST's) in the generalized incomplete hypercubes. An edge-disjoint spanning tree (EDST) is a directed spanning tree in which the edges in the tree are distinct. A directed MUST is composed of at least one EDST, in which each directed subtree along with the edges from the root of the subtree to the root of MUST forms an EDST. The EDST's in a MUST are all edge-disjoint in the sense of directed edges.

The MUST in the hypercube $Q_n$, denoted as $MUST_{Q_n}$, has height $n + 1$ and consists of $n$ number of EDST's [47]. The $i$th EDST, where $0 \leq i \leq n - 1$, in the MUST$_{Q_n}$ is constructed as follows. First, the root $v_0$ is linked to its $i$th child $r_i$; having address $0^{[i+1]} = (0^{n-i-1}10^i)$ through an $i$-dimensional directed edge. Let $v_j$ be a node
in the $i$th $EDST$ rooted at $r_i$. Node $v_j$ differ from its parent $v_k$ at the $(m+1)$th bit, i.e. $BR(k \oplus j) = (0^{n-m-1}10^m)$. Then $v_j$ is linked to its children through the edges of dimension $(m+1) \mod n, (m+2) \mod n, \ldots, (m-1) \mod n$.

By this consistent children-parent relation, the embedded $MUST_Q$ rooted at node $v_0$ is uniquely defined by the dimensions of edges, and works as a template. The $MUST_Q$ rooted at an arbitrary source node $s$ can be simply translated from the $MUST_Q$ rooted at $v_0$. The address of a node $v_j$ in the $MUST_Q$ rooted at $s$ is obtained by a bitwise Exclusive-OR operation on the address of its corresponding node $v_i$ in the $MUST_Q$ rooted at $v_0$ with the address of $s$. Figure 6.3 shows the $MUST_Q$. In this figure, the nodes marked by $r_i$ for $1 \leq i \leq 4$ are the roots of four $EDST$'s. The directed edges $< s, r_i >$ indicated by the broken arcs are the reversed edges. The subtree rooted at $r_i$ associated with the directed edge $< r_i, s >$ forms an $EDST$ of $Q_4$. The part of the $MUST_Q$ circled by the broken lines is the the first $EDST$ of $Q_4$.

Now consider the generalized incomplete hypercube network $GIQ(N)$, where
$2^{k-1} < N < 2^k$ and $k > 1$. Let $N = \sum_{i=1}^{l} 2^{p_i}$, where $1 \leq i \leq k$ and $p_{i+1} > p_i$; and $GIQ(N) = \{Q_{p_1}, Q_{p_{r-1}}, \ldots, Q_{p_1}\}$. The $l$ node-sets of subgraphs are $V_{p_1} = \{v_\alpha \mid 0 \leq \alpha < 2^{p_1}\}$ and $V_{p_1}^{(l-m)} = \{v_\alpha \mid \sum_{j=m+1}^{l} 2^{p_j} \leq \alpha < \sum_{j=m}^{l} 2^{p_j}\}$ for $1 \leq m \leq (l-1)$.

Assume that the source node $s_i \in V_{p_1}$. The following definition provides a mapping from $s_i$ to the node $s'_j \in V_{p_j}$, called the virtual source node, where $i \neq j$. Recall that a node $i^{[l]} = (b_k b_{k-1} \ldots b_j \ldots b_1)$, in which the $j$th bit of $i$ is complemented.

**Definition 6.1:** Let $N = \sum_{i=1}^{l} 2^{p_i}$, where $2^{k-1} < N < 2^k$, $1 \leq i \leq k$ and $p_{i+1} > p_i$.

Assume that the binary representation of address $s_i$ is $BR(s_i) = (1^{k-1-p_i} b_{p_i} b_{p_{i-1}} \ldots b_1)$, where $b_\alpha \in \{0, 1\}$ for some $\alpha$. A virtual address of $s'_j$ for $j \neq i$ is determined as follows:

**(M1):** If $i < j$ then $s'_j = s_i^{[p_j+1]}$.

**(M2):** If $i > j$ then consider the following two cases:

**(M2.1)** When all the powers $p$'s in $N$ form a sequence of consecutive numbers,

$$BR(s'_j) = (1^{k-1-p_j} 0 b_{p_j} b_{p_{j-1}} \ldots b_1),$$

where $(b_{p_j} b_{p_{j-1}} \ldots b_1)$ are the rightmost $p_j$ bits in $BR(s_i) = (b_{p_i} b_{p_{i-1}} \ldots b_1)$.

**(M2.2)** Otherwise, start from the right most significant $p_i$ to find the first $p_f$ which is not continuous in the sequence of $p$'s, i.e. $p_f \neq p_{(f+1)} - 1$ then

1. $BR(s'_j) = (1^{k-1-p_j} 0 b_{p_j} b_{p_{j-1}} \ldots b_1)$ for $f < j < i$;
2. $BR(s'_j) = (b_k' b'_{k-1} \ldots b'_{p_{(j+1)}+2} 10^{p_{(f+1)}})$ for $1 \leq j \leq f$,

where $(b_k' b'_{k-1} \ldots b'_{p_{(j+1)}+2})$ is the substring of $BR(s'_j) = (b'_k b'_{k-1} \ldots b'_1)$, the address of $s'_{j+1}$.

**Example 6.6:** Consider $GIQ(N)$, for $N = 11$ and 27. Let the source nodes are $s_i = v_6$ and $v_{22}$ in $GIQ(11)$ and $GIQ(27)$, respectively.

(1) When $N = 11 = 2^3 + 2^1 + 2^0$, $p_3 = 3$, $p_2 = 1$, and $p_1 = 0$. $BR(s_i) = BR(6) = (0110) \in V^3$. Thus, $p_i$'s do not form a sequence of consecutive numbers, and $p_f = p_2 = 1 \neq p_{(f+1)} - 1 = p_3 - 1$. By Rule (M2.2), $BR(s'_4) = (1000) \in V^1$ and $BR(s'_4) =$
Figure 6.4: The (virtual) source nodes in $GIQ(11)$ and $GIQ(27)$. 
(1010) ∈ V°, as shown in Figure 6.4(a).

(2) When \( N = 27 = 2^4 + 2^3 + 2^1 + 2^0 \), \( p_4 = 4, p_3 = 3, p_2 = 1, \) and \( p_1 = 0 \) and \( BR(s_i) = BR(22) = (10110) \in V^3 \). By Rule (M1), \( s_4' = v_6 \in V^4 \). Clearly, \( p_f = p_2 \) is the first \( p \) that is discontinuous. By (M2.2), \( s_2' = v_{24} \in V^1 \) and \( s_1' = v_{26} \in V^0 \), respectively. Figure 6.4(b) shows these (virtual) source nodes.

Consider \( GIQ(N) = \{Q_{p_1}, Q_{p_1-1}, \ldots, Q_{p_i}\} \). By Definition 6.1, the virtual source node (other than the source node) in each induced subgraph can be determined. Since each induced subgraph forms a subcube, we can construct the \( MUST_{Q_{p_j}} \) rooted at the (virtual) source node in a subgraph \( Q_{p_j} \), consisting of \( p_j \) EDST's, for \( 1 \leq j \leq l \). Then these \( MUST \)'s in subgraphs are combined step by step to construct the \( MUST \) of \( GIQ(N) \), denoted as \( MUST_{GIQ(N)} \).

Assume that the source node \( s_i \) is in \( Q_{p_i} \). The construction of \( MUST_{GIQ(N)} \) consists of two phases. The first phase unites the \( MUST \)'s in subgraphs \( \{Q_{p_i}, Q_{p_i-1}, \ldots, Q_{p_1}\} \), while the second phase unites the \( MUST \)'s in the remaining subgraphs \( \{Q_{p_1}, Q_{p_1-1}, \ldots, Q_{p_{i+1}}\} \). We will use the notation \( MUST_\beta \) to indicate the temporary \( MUST \) which is united from the \( MUST \)'s of \( \{Q_\beta, Q_{\beta-1}, \ldots, Q_\alpha\} \).

At the beginning, \( MUST_{GIQ(N)} = MUST_{p_1} \) rooted at the virtual source node \( s_{p_1}' \), which contains \( p_1 \) number of EDST's, indicated by \( E_i \), rooted at nodes \( r_i \), where \( 1 \leq i \leq p_1 \). While the \( MUST_{p_2} \) in the subgraph \( Q_{p_2} \) rooted at the virtual source node \( s_{p_2}' \) has \( p_2 \) number of EDST's, \( E_j \)'s, rooted at \( r_j' \), where \( 1 \leq j \leq p_2 \). To construct \( MUST_{p_1} \), we first append \( E_{(p_1-j)} \) to \( E'_{(p_2-j-1)} \), by making \( r_{(p_1-j)} \) the child of \( r_{(p_2-j)}' \) in \( E'_{(p_2-j-1)} \) through the \( p_2 \)-dimensional edges, where \( 0 \leq j \leq p_1 - 1 \). Note that node \( r_{(p_1-j)}' \) is obtained from \( r_{(p_1-j)} \) by complementing the \( p_2 \)th bit counting from the least significant bit as one.

Let \( T_{p_1} \) be a two-level tree rooted at the node \( s_{p_1}' \) (the root of \( MUST_{p_1} \) ) having
$p_1$ children denoted as $r_1, r_2, \ldots, r_{p_1}$. Append the $T_{p_1}$ tree to $E'_{p_2}$ by making $s'_{p_1}$ the child of $s'_{p_2}$ through a $p_2$-dimensional edge. For the remaining nodes of $Q_{p_1}$ which are not used in $T_{p_1}$, i.e. in $V_{P_1} - V(T_{P_1})$, we link them to $E'_{p_2}$ through the $p_2$-dimensional edges. In this step, the $p_2$th EDST of $\text{MUST}_{p_1}^{\text{P_2}}$ is constructed and some nodes will probably be connected to the root $s'_{p_2}$ of $\text{MUST}_{p_1}^{\text{P_2}}$, called the stray nodes. The following procedure $\text{UNION MUST PHASE 1()}$ unites $\text{MUST}_{P_1}^{p_m}$ with $\text{MUST}_{P_{m+1}}^{p_{m+1}}$ to construct $\text{MUST}_{p_1}^{p_{m+1}}$ for $1 \leq m \leq i - 1$. Repeating this procedure, we can construct $\text{MUST}_{p_1}^{p_2}, \text{MUST}_{p_1}^{p_3}, \ldots, \text{MUST}_{p_1}^{p_i}$.

procedure $\text{UNION MUST PHASE 1(} \text{MUST}_{p_1}^{p_m}, \text{MUST}_{p_{m+1}}^{p_{m+1}}, \text{MUST}_{p_1}^{p_{m+1}} \text{)}$

begin

1. Assume that the roots of $\text{MUST}_{p_1}^{p_m}$ and $\text{MUST}_{p_{m+1}}^{p_{m+1}}$ are $s_{p_m}$ and $s_{p_{m+1}}$, respectively.

2. Let $p_m$ number of EDST's of $\text{MUST}_{p_1}^{p_m}$ be $E_{(p_m-j)}$ rooted at $r_{(p_m-j)}$ and let $p_{m+1}$ number of EDST's of $\text{MUST}_{p_{m+1}}^{p_{m+1}}$ be $E'_{(p_{m+1}-j')}$ rooted at $r'_{(p_{m+1}-j')}$, where $0 \leq j \leq p_m - 1$ and $0 \leq j' \leq p_{m+1} - 1$.

3. Append $E_{(p_m-j)}$ to $E'_{(p_{m+1}-j-1)}$ by making $r_{(p_m-j)}$ the child of $r'_{(p_{m+1}-j)}$ in $E'_{(p_{m+1}-j-1)}$.

4. Construct $T_{p_m}$ which is a two-level tree rooted at $s_{p_m}$ and has $p_m$ children, given by $r_{(p_m-j)}$, for $0 \leq j \leq p_m - 1$.

5. Append $T_{p_m}$ to $E'_{p_{m+1}}$ by making $s_{p_m}$ the child of $s_{p_{m+1}}$ in $E'_{p_{m+1}}$.

6. Let $V(T_{p_m}) = \{s_{p_m}$ and $r_{(p_m-j)}$, for $0 \leq j \leq p_m - 1\}$, and link each node $v_i \in (V_{p_m} - V(T_{p_m}))$ to $v'_{p_{m+1}}$ in $E'_{p_{m+1}}$.

7. $\text{MUST}_{p_1}^{p_{m+1}} \leftarrow \text{MUST}_{p_{m+1}}^{p_{m+1}}$.

end.

In the second phase, $\text{MUST}_{p_1}^{p_i}$ is extended to unite the $\text{MUST}$'s of the subgraphs $\{Q_{p_1}, Q_{p_{i-1}}, \ldots, Q_{p_i}\}$. Procedure $\text{UNION MUST PHASE 2()}$ shows how to unite $\text{MUST}_{p_1}^{p_m}$ with $\text{MUST}_{p_{m+1}}^{p_{m+1}}$ to construct $\text{MUST}_{p_1}^{p_{m+1}}$ for $i \leq m \leq i - 1$, by appending
procedure UNION_MUST_PHASE_2(MUST_{p_1}^{p_m}, MUST_{p_{m+1}}^{p_m+1}, MUST_{p_1}^{p_{m+1}})
begin
1 Assume that the roots of MUST_{p_1}^{p_m} and MUST_{p_{m+1}}^{p_m+1} are s_{p_i} and s'_{p_{m+1}}, respectively.
2 Let p_m number of EDST's of MUST_{p_1}^{p_m} be E_j rooted at r_j and let p_{m+1} number of EDST's of MUST_{p_{m+1}}^{p_m+1} be E'_j rooted at r'_j, where 1 \leq j \leq p_m and 1 \leq j' \leq p_{m+1}.
3 Append E'_j to E_j by making r'_j the child of r_{p_{m+1}} in E'_j.
4 Let U_{p_1}^{p_m} = V_{p_1} \cup V_{p_2} \ldots \cup V_{p_m}.
5 Create a new EDST by reversing the edge < s_{p_{m+1}}, s'_{p_{m+1}} > in E_{p_{m+1}}',
   and link s_{p_{m+1}} to s'_{p_{m+1}} through the p_{m+1}-dimensional edge. Also make each node
   v_t \in (U_{p_1}^{p_m} - s_{p_i}) the child of v_{t|p_{m+1}} in E_{p_{m+1}}'.
6 MUST_{p_{m+1}} \leftarrow MUST_{p_1}^{p_m}.
end.

Based on the preceding two phases, the construction of the MUST of GIQ(N) is
given as follows.

Algorithm MUST_GIQ: Construct the MUST_{GIQ(N)}, where 2^{k-1} < N < 2^k.
begin
1 Let GIQ(N) = \{Q_{p_i}, Q_{p_{i-1}}, \ldots, Q_{p_1}\}, where N = \Sigma_{i=1}^{l} 2^{p_i}, 1 \leq l \leq k and
   p_{i+1} > p_i.
2 Assume that the source node s_i is in Q_{p_i}.
3 By Definition 6.1, compute the (virtual) source nodes in each subgraph.
4 For each subgraph, construct the MUST rooted at its corresponding (virtual)
   source node.
5 for j = 1 to i - 1 do { PHASE 1}
   call UNION_MUST_PHASE_1(MUST_{p_1}^{p_j}, MUST_{p_{j+1}}^{p_j+1}, MUST_{p_1}^{p_{j+1}}).
for \( j = i \) to \( l - 1 \) do \{ PHASE 2\}

\[
\text{call UNION.MUST.PHASE.2}(MUST_{p_i}^j, MUST_{p_i+1}^{p_m+1}, MUST_{p_i+1}^{p_m+1}).
\]
end.

**Theorem 6.2.** The multiple spanning tree \( MUST_{GIQ(N)} \), where \( 2^{k-1} < N < 2^k \), is composed of \( \phi_{min} \) edge-disjoint spanning trees, where \( \phi_{min} \) is the minimum degree of \( GIQ(N) \).

**Proof:** It can be easily shown that \( MUST_{GIQ(N)} \) is edge-disjoint by induction on \( p_m \) of \( MUST_{p_i}^{p_m} \) for \( 1 \leq m \leq l \). Thus, the remaining part needed to prove is that the union procedures construct the edge-disjoint multiple spanning tree \( MUST_{p_i}^{p_m+1} \) from \( MUST_{p_i}^{p_m} \) and \( MUST_{p_i}^{p_m+1} \). Let \( GIQ(N) = \{ Q_{p_1}, Q_{p_2-1}, \ldots, Q_{p_l} \} \). According to Definition 6.1, the \( MUST \)'s rooted at their corresponding (virtual) source nodes can be constructed in each subgraph. Note that each \( MUST_{p_i}^{p_m} \) consists of \( p_m \) edge-disjoint trees, spanning the nodes of \( Q_{p_m} \) [47].

In Phase 1, \( p_m \) number of EDST's of \( MUST_{p_i}^{p_m} \), denoted as \( E_{(p_m-j)} \), are appended to their corresponding \( p_m \) number of EDST's of \( MUST_{p_i}^{p_m+1} \), denoted as \( E_{(p_m+1-j)} \). For the \((p_m+1)\)th EDST of \( MUST_{p_i}^{p_m+1} \), denoted as \( E_{(p_m+1)} \), the \( T_{p_m} \) is constructed in which the tree edges are not used in any EDST's in \( MUST_{p_i}^{p_m+1} \), and is appended to \( E_{(p_m+1)} \). The remaining nodes \( v_1 \) in \( U_{p_i}^{p_m} \) but not used in \( T_{p_m} \) are linked to \( v_{(p_m+1)} \) in \( E_{(p_m+1)} \), where \( U_{p_i}^{p_m} = \cup_{j=1}^{p_m} V_{p_i} \). In Phase 2, \( p_m \) number of EDST's of \( MUST_{p_i}^{p_m+1} \) are appended to their corresponding EDST's of \( MUST_{p_i}^{p_m} \). We also create a new EDST to \( MUST_{p_i}^{p_m} \) by reversing the edge \( < r_{(p_m+1)}', s_{(p_m+1)}' > \) in the \((p_m+1)\)th EDST of \( MUST_{p_i}^{p_m+1} \), \( E_{(p_m+1)}' \), where \( r_{(p_m+1)}' \) is the root of \( E_{(p_m+1)}' \) and \( s_{(p_m+1)}' \) is the virtual source node of \( Q_{p_m+1} \). The nodes \( v_1 \in (U_{p_i}^{p_m} - s_{p_i}) \) are linked to \( v_{(p_m+1)} \) in \( E_{(p_m+1)}' \), where \( s_{p_i} \) is the source node. So, the \( MUST_{p_i}^{p_m+1} \) is edge-disjoint.

Now, we want to prove that \( MUST_{GIQ(N)} \) has \( \phi_{min} \) edge-disjoint spanning
trees. It is known that \( 1 \leq \phi_{\text{min}} \leq \log N \) in GIQ(N). Assume that \( s_i \) is in \( Q_{p_i} \). By Algorithm MUST_GIQ, we construct the MUST\(_{GIQ}(N)\) rooted at \( s_i \). Let \( DEG(s_i; N) = x \). Among the \( x \) neighbors of \( s_i \), only \( p_i \) of them are in \( Q_{p_i} \), linked by the edges of dimensions 0, 1, \ldots, \( p_i - 1 \) and the other \( (x - p_i) \) neighbors are in \( Q_{p_j} \) for \( j \neq i \). If \( DEG(s_i; N) = \phi_{\text{min}} \), then \( (x - p_i) \) neighbors, \( s_i^{[j]} \), are in \( Q_{p_j} \) for \( j > i \) through \( j \)-dimensional edges. Let us consider the following two cases.

**Case 1:** \( DEG(s_i; N) = \phi_{\text{min}} \).

In Phase 1, \( MUST_{p_i}^\text{t} \) rooted at \( s_i \) consisting of \( p_i \) EDST's is constructed, and in Phase 2, \( (x - p_i) \) new EDST's are created (Line 4-5 in \textsc{Union-MUST-Phase.2}) and appended to \( s_i \). So, there are \( \phi_{\text{min}} \) edge-disjoint spanning trees.

**Case 2:** \( DEG(s_i; N) > \phi_{\text{min}} \).

Assume that \( v_{p_y} \) is a node with \( \phi_{\text{min}} \) and is in \( Q_{p_y} \) for \( y < i \). Thus, \( v_{p_y} \) is linked to \( p_y \) neighbors in \( Q_{p_y} \) and \( (\phi_{\text{min}} - p_y) \) neighbors in \( Q_{p_j} \) for \( j > y \). Note that \( MUST_{p_i}^\text{t} \) has \( p_i \) EDST's. According to \textsc{Union-MUST-Phase.1}, \( (p_y + 1) \) EDST's of \( MUST_{p_i}^\text{t-1} \) will be created and appended to the \( (p_i - j) \) EDST of \( MUST_{p_i}^\text{t} \), where \( 0 \leq j \leq p_y \). Thus, the \( (p_y + 1) \) edge-disjoint trees of \( MUST_{p_i}^\text{t} \) are the spanning trees of \( \{Q_{p_i}, Q_{p_i-1}, \ldots, Q_{p_1}\} \). In Phase 2, \( (\phi_{\text{min}} - p_y - 1) \) additional EDST's are added. Hence, there is a total of \( \phi_{\text{min}} \) EDST's in \( MUST_{GIQ}(N) \) which are the \( (p_i - j) \)th edge-disjoint trees \( E_{(p_i-j)} \) in \( MUST_{GIQ}(N) \), where \( 0 \leq j < \phi_{\text{min}} \).

**Theorem 6.3.** The height \( h \) of \( MUST_{GIQ}(N) \) satisfies \( \lceil \log N \rceil + 1 \leq h \leq 3 \lceil \log N \rceil - 2 \).

**Proof:** Let \( GIQ(N) = \{Q_{p_1}, Q_{p_1-1}, \ldots, Q_{p_1}\} \) and \( s_i \) be in \( Q_{p_i} \). Consider the two extreme cases that \( s_i \in Q_{p_1} \) and \( s_i \in Q_{p_l} \). When \( s_i \in Q_{p_1} \), by Definition 6.1, the virtual source node \( s_j \in Q_{p_j} \) for \( 1 < j \leq l \), has Hamming distance \( \rho(s_i, s_j) = 1 \). By Algorithm MUST_GIQ, the height of \( MUST_{GIQ}(N) \) is \( h = p_l + 2 = \lceil \log N \rceil + 1 \).
Consider another extreme case when \( s_i \in Q_{p_l} \). Let \( N = 2^k - 1 \) and \( s_i = v_{2^{k-1}} \). By Rule (M2.1), the virtual source node \( s_j \) in \( Q_{p_j} \) for \( 1 \leq j < l \), has \( \rho(s_i, s_j) \leq 2 \). It implies that appending the EDST's of \( \text{MUST}^{p_j}_i \) to those of \( \text{MUST}^{p_{j+1}}_{i+1} \) will increase the height of \( \text{MUST}_{G\text{IQ}(N)} \) by two. Thus, the height of \( \text{MUST}_{G\text{IQ}(N)} \) for this case is \( 3(p_l - 1) + 1 = 3 \lceil \log N \rceil - 2 \). \( \square \)

Figure 6.6(a)–(c) shows \( Q_0, Q_1 \), and \( \text{MUST}_0 \). Figure 6.6(d) shows the \( \text{MUST}_{G\text{IQ}(N)} \) rooted at node \( v_{22} \). The virtual nodes determined by Definition 6.1 are provided in Part (2) of Example 6.6. Note that the two EDST's of \( \text{MUST}_0 \) are appended to the second and third EDST's of \( \text{MUST}_{G\text{IQ}(27)} \) indicated as broken lines in Figure 6.6(d). The four triangular shapes denote the four EDST's in \( \text{MUST}_4 \) rooted at node \( v_6 \) as shown in Figure 6.5. When the minimum node-degree \( \phi_{\text{min}} = 1 \), \( \text{MUST}_{G\text{IQ}(N)} \) has only one EDST. In this case, \( \text{MUST}_{G\text{IQ}(N)} \) is degraded to the spanning tree of \( G\text{IQ}(N) \). Therefore, we are interested in \( \text{MUST}_{G\text{IQ}(N)} \) when \( \phi_{\text{min}} \geq 2 \). By Theorems 6.2 and 6.3, the total time for one-port communication is \( M + h - 1 \) and for all-port
Figure 6.6: The $MUST_{GQ(27)}$ rooted at node $v_{22}$. 
communication is \( \left\lceil \frac{M}{\phi_{\text{min}}} \right\rceil + h - 1 \), where \( \log N \leq h \leq 3\log N - 2 \).

6.3. Summary

The topology of generalized incomplete hypercubes provides a fundamental structure for the enhanced hypercube-family networks. In this chapter, we have constructed two types of broadcasting primitives, spanning trees \( ST_{GIQ(N)} \) and multiple spanning trees \( MUST_{GIQ(N)} \), for one-port and all-port communication models. The height of \( ST_{GIQ(N)} \) is at most the diameter \( \log N \), and the \( MUST_{GIQ(N)} \) is optimally composed of \( \phi_{\text{min}} \) edge-disjoint spanning trees, where \( \phi_{\text{min}} \) is the minimum degree of \( GIQ(N) \). The technique that constructs the \( MUST_{GIQ(N)} \)'s in generalized incomplete hypercubes can be modified to construct the \( MUST \)'s in enhanced generalized incomplete hypercubes and Hamming cubes shown in the following chapters.
CHAPTER 7

BROADCASTING TREES IN HAMMING CUBES

In this chapter, we design the broadcasting schemes for Hamming cubes by constructing the embedded broadcasting trees, the spanning trees and multiple spanning trees. The construction of the embedded broadcasting trees in the Hamming cubes depends on the size of the network and the identification of the source node. The spanning tree $ST_{HC_n}$ in the $n$-dimensional Hamming cube ($HC_n$) is constructed from a newly derived tree, called the binomial plus spanning tree $B^+ST$, which is a variant of the binomial spanning tree $BST$ [47]. With the help of such trees as $ST_{HC_n}$, $B^+ST$, and $BST$, the embedded spanning tree $ST_{HC(N)}$ in the Hamming cube $HC(N)$ of arbitrary order $N$, where $2^{k-1} < N < 2^k$ and $k > 1$, can be further constructed.

Since the folded hypercube is a spanning subgraph of $HC_n$, the exiting multiple spanning tree in the folded hypercube [36] also provides an embedding of the multiple spanning tree $MUST_{HC_n}$ in $HC_n$. Thus, we will only concentrate on the construction of the multiple spanning tree $MUST_{HC(N)}$ in $HC(N)$. For this purpose, the technique presented in Chapter 6 which constructs the multiple spanning tree $MUST_{GIQ(N)}$ in generalized incomplete hypercube can be used here [26].

The following section first discusses the construction of spanning trees.

7.1. Construction of Spanning Trees $ST_{HC_n}$ in $HC_n$, where $n > 1$

A directed spanning tree is edge-disjoint if its tree edges are disjoint. The embedded spanning trees of Hamming cubes are directed and satisfy the property of edge-disjointness. The constructions of $ST_{HC}$'s are divided into two cases depending
on the sizes of Hamming cubes: (1) a power of two and (2) an arbitrary (not a power of two) number of nodes.

7.1.1. Binomial Plus Spanning Trees ($B^+ST$'s)

We can generalize the binomial spanning tree ($BST$) structure which is defined in Section 6.3.1 by considering only the first few levels from the root. The new generalized tree forms an incomplete $BST$ defined next.

**Definition 7.1:** An incomplete binomial spanning tree of $\alpha$ levels, indicated as $BST_{\alpha}^l$, with respect to an $l$-level $BST$ is a binomial tree consisting of the first $\alpha$-levels from the root.

The decomposition of $n$-dimensional Hamming cube yields $HC_n = \{HC_2, Q_2, Q_3, \ldots, Q_{n-1}\}$, where $n \geq 2$. The induced subgraphs are specified by the node-sets $V^1 = \{v_0, v_1, v_2, v_3\}$ and $V^i = \{v_j \mid 2^i \leq j < 2^{i+1}\}$, for $2 \leq i \leq n-1$, respectively. Assume that the node $s \in V^{i-1}$ corresponding to the subgraph $Q_{t-1}$, when $3 \leq t \leq n$, and $HC_2$ if $t = 2$. By the recursive decomposition, $HC_n = \{HC_2, Q_2, Q_3, \ldots, Q_{n-1}\} = \{HC_t, Q_t, \ldots, Q_{n-1}\}$. Recall that the subgraphs $\{HC_2, Q_2, Q_3, \ldots, Q_{t-1}\}$ form a $t$-dimensional Hamming cube $HC_t$.

The construction of the spanning tree $ST_{HC_n}$ in $HC_n$ consists of two phases. In the first phase, we construct the spanning tree rooted at the source node $s$, called the binomial plus spanning tree ($B^+ST$), for the subgraph $HC_t$. Then, we grow the $B^+ST$ greedily to span the remaining nodes belonging to the subgraphs $\{Q_t, Q_{t+1}, \ldots, Q_{n-1}\}$.

The $B^+ST$ rooted at node $s$ is constructed as follows. We first construct two incomplete binomial spanning trees $BST_{\alpha}^l$ and $BST_{\beta}^l$ rooted at nodes $v_0$, where $\alpha = \beta = \lfloor \frac{t}{2} \rfloor$ if $t$ is odd; otherwise $\alpha = \beta + 1 = \frac{t}{2}$. Then translate the incomplete $BST_{\beta}^l$
Figure 7.1: The $B^+ST$'s in $HC_4$ and $HC_5$

rooted at node $v_0$ to the $BST_0^j$ rooted at node $v_0$. Clearly $v_0 = v_{2^n-1}$. Thus, we can add an $n_t$-dimensional E2-edge between roots $v_0$ and $v_0$ and make $v_0$ the new root. The new tree also works as a template to form a $B^+ST$ rooted at an arbitrary node. By this template, the $B^+ST$ rooted at node $s$ can be translated from the $B^+ST$ rooted at node $v_0$. Note that the incomplete $BST_0^j$ in the $B^+ST$ will be translated twice, corresponding to node $v_0$ at the first time and to node $s$ at the second time. Therefore, one can verify that after these two translations (bitwise Exclusive-OR operations), the root of the $BST_0^j$ is node $s$. Figures 7.1(a) and (b) show two $B^+ST$'s in $HC_4$ and $HC_5$, respectively. The labels on the edges are the dimensions of edges in the networks $HC_4$ and $HC_5$.

By the construction of the $B^+ST$ in $HC_t$, it can be seen that the height of $B^+ST$ is $\lceil \frac{t}{2} \rceil$ bounded by the diameter of $HC_t$. The following property gives the number of nodes at different levels of a $B^+ST$.

**Property 7.1.** Let $n_i$ be the number of nodes at level $i$ of the embedded $B^+ST$ in $HC_t$ for $t \geq 2$. If $t$ is even, then $n_i = \binom{t+1}{i/2}$ for $0 \leq i \leq \lceil \frac{t}{2} \rceil$, otherwise
98

\[
n_i = \begin{cases} 
\binom{i+1}{i} & \text{for } 0 \leq i \leq \left\lfloor \frac{t}{2} \right\rfloor, \\
\frac{1}{2}\binom{i+1}{i} & \text{for } i = \left\lceil \frac{t}{2} \right\rceil.
\end{cases}
\]

Proof: Recall that the embedded \(B^+ ST\) in network \(HC_t\) is constructed from two incomplete binomial spanning trees, \(BST^t_\alpha\) and \(BST^t_\beta\). Let us first consider the case when \(t\) is even. At level 0, the \(B^+ ST\) has \(\binom{t}{0}\) nodes and at level \(i\) for \(1 \leq i \leq \frac{t}{2}\), the \(B^+ ST\) has \(\binom{i}{i} + \binom{i-1}{i-1}\) nodes, obtained from the number of nodes at \(i\)th level of \(BST^t_\alpha\) and at \((i-1)\)th level of \(BST^t_\beta\), respectively. By Pascal Identity, \(\binom{i}{i} + \binom{i-1}{i-1} = \binom{i+1}{i}\).

When \(t\) is odd, \(BST^t_\alpha\) has no node at level \(\left\lfloor \frac{t}{2} \right\rfloor\). Hence, the property follows. □

Lemma 7.1. The \(B^+ ST\) in \(HC_t\) is the spanning tree of \(HC_t\).

Proof: The embedded \(B^+ ST\) rooted at node \(s\) in \(HC_t\) is composed of two incomplete binomial trees, \(BST^t_\alpha\) and \(BST^t_\beta\). The \(BST^t_\alpha\) consists of the nodes \(v_i\) in \(HC_t\) with the Hamming distance \(\rho(s, v_i) \leq \left\lfloor \frac{t}{2} \right\rfloor\), and the \(BST^t_\beta\) contains the nodes \(v_j\) in \(HC_t\) with \(\rho(s, v_j) > \left\lceil \frac{t}{2} \right\rceil\). Hence, the lemma follows. □.

We now describe the construction of the spanning tree \(ST_{HC_n}\) for the second phase, that is to grow the \(B^+ ST\) greedily to span the nodes in the subgraphs \(\{Q_t, Q_{t+1}, \ldots, Q_{n-1}\}\). Consider the decomposition \(HC_n = \{HC_t, Q_t, \ldots, Q_{n-1}\}\). Assume that we have spanned the nodes in the subgraph \(Q_t\). Then the next subgraph to be included is \(Q_{t+1}\). The induced node-set for subgraph \(Q_{t+1}\) is \(V^{t+1} = \{v_i \mid 2^t \leq i < 2^{t+1}\}\). Let \(V^{t+1} = V_1 \cup V_2\) such that \(V_1 = \{v_i \mid 2^t \leq i < 2^{t+1} + 2^t\}\) and \(V_2 = \{v_i \mid 2^t + 2^t \leq i < 2^{t+2}\}\). By the definition of Hamming cubes, each node \(v_i \in V_1\) is connected to \(v_{i+2^t}\) in \(HC_t\) through a \((t+1)\)-dimensional \(E_1\)-edge, and \(v_i \in V_2\) is connected to \(v_{i+2^t}\) in \(HC_t\) through an \(n_{t+2^t}\)-dimensional \(E_2\)-edge. By this property, the nodes in subgraph \(Q_{t+1}\) can be spanned by growing the subgraph \(HC_t\). After spanning subgraph \(Q_{t+1}\), the recursive decomposition becomes
$HC_n = \{HC_{t+1}, Q_{t+1}, \ldots, Q_{n-1}\}$. Using the same method, the nodes in subgraph $Q_{t+2}$ can be spanned by the nodes in $HC_{t+1}$ through $(t + 2)$-dimensional $E1$-edges and $n_{t+3}$-dimensional $E2$-edges. The same process is repeated until the subgraph $Q_{n-1}$ is spanned.

Depending on the root $s$ of the $B^+ST$, the inclusion of subgraph $Q_t$ has to be specially dealt with. When $t = 2$, node $s \in HC_2$. We first identify whether $s$ belongs to $\{v_0, v_1\}$ or $\{v_2, v_3\}$. The nodes in $Q_2$ are then spanned by making them the children of the nodes in the selected through the $t$-dimensional $E1$-edges and $n_{t+1}$-dimensional $E2$-edges. When $t \neq 2$, node $s \in Q_{t-1}$. Taking advantage of $t$-dimensional $E2$-edges, the nodes $v_\gamma$ in $Q_{t-1}$ with the Hamming distance $\rho(s, v_\gamma) > \lceil \frac{t}{2} \rceil$ are replaced by nodes $v_\delta$ such that $v_\delta = v_\gamma t_0$. Let $V$ be the new node-set of $Q_{t-1}$. Then the nodes in $Q_t$ can also be spanned by making them the children of the nodes in $V$ through the $t$-dimensional $E1$-edges and $n_{t+1}$-dimensional $E2$-edges. The entire algorithm is formally described below.

**Algorithm ST$_{HC_n}$:** Construct the spanning tree of network $HC_n$.

**Input:** the source node $s$. **Output:** the $ST_{HC_n}$ rooted at $s$.

**begin**

1. Assume that node $s \in V^{t-1}$.

2. if $(t = 2)$ then $s \in HC_2$ else $s \in Q_{t-1}$.

3. Construct the $B^+ST$ for $HC_t$ and let it be $ST_{HC}$.

4. if $(t = n)$ then return $ST_{HC}$.

else begin

4.1. if $(t = 2)$ then

begin

if $s \in \{v_0, v_1\}$ then $V = \{v_0, v_1\}$.
else $V = \{v_2, v_3\}$.

end

4.2 else begin

4.2.1 Let $V'$ be the set of nodes $v_r$ such that $v_r \in V^{t-1}$ and $\rho(s, v_r) > \lceil \frac{t}{2} \rceil$.

4.2.2 Let $V''$ be the set of nodes $v_r$ such that $v_r \in V^{t-1}$ and $v_r = v_{s(t)}$.

4.2.3 $V \leftarrow V^{(t-1)} - V' + V''$.

end

4.3 for each node $v_i \in V$ do

4.3.1 make two nodes $v_{(t+1)}$ and $v_{(t+1)}$ the children of $v_i$.

4.4 for $j = t + 1$ to $n - 1$ do

4.4.1 For each node $v_i \in V'$, where $V' = \{v_0, v_1, \ldots, v_{2(t-1)}\}$, make two nodes $v_{(j+1)}$ and $v_{(j+1)}$ the children of $v_i$.

end

end.

Figure 7.2 shows the spanning tree, $ST_{HC_5}$, rooted at node $v_{10}$ in $HC_5 = \{HC_2, Q_2, Q_3, Q_4\}$. The source $v_{10} \in Q_3$. The $B^+ST$ in $HC_4$ rooted at $v_{10}$ is shown by the dark lines and the nodes $v_i$ in $Q_4$ for $16 \leq i < 32$, are spanned through 4-dimensional and $n_5$-dimensional edges (shown by broken lines). Note that node $v_{13}$ is replaced by $v_2$ when including $Q_4$.

According to Algorithm $ST_{HC_n}$, the height of the growing $B^+ST$ will increase by one when including every two subgraphs $Q_i$ and $Q_{i+1}$, where $i = t + 1, t + 3, \ldots$, if $t$ is odd and $i = t, t + 2, \ldots$, otherwise. Therefore, the increased height of the $B^+ST$ is $\lceil \frac{n-1}{2} \rceil$. Due to the height $\lceil \frac{n}{2} \rceil$ of the $B^+ST$ in $HC_t$, we obtain

Property 7.2. The height of the $ST_{HC_n}$ for $n > 1$ is $\lceil \frac{n}{2} \rceil$.

Lemma 7.2. The $ST_{HC_n}$ in $HC_n$ is the spanning tree of $HC_n$ for $n > 1$. 
Figure 7.2: The $ST_{HC_n}$ rooted at node $v_{10}$

**Proof:** By Lemma 7.1, the $B^+ST$ is the spanning tree of $HC_t$, and the $B^+ST$ grows to span the nodes in the subgraphs $Q_i$ for $t \leq i \leq n - 1$ through $i$-dimensional $E1$-edges and $n_{i+1}$-dimensional $E2$-edges. Hence, the lemma follows. □

Clearly, the maximum degree $\phi_{\text{max}}(ST_{HC_n})$ of the spanning tree $ST_{HC_n}$ is the degree $\text{DEG}(s; 2^n)$ of the source node $s$. Also the minimum node-degree of $HC_n$ is $n + 1$. Therefore, the degree $\text{DEG}(s; 2^n) \geq n + 1$ which is also greater than $\lceil \frac{1}{2} \rceil$, the height of $ST_{HC_n}$. Thus, the significant value $\text{DEG}(s; 2^n)$ is used to compute the time period which is the waiting time required to send a packet through a same edge (a communication port) in the one-port communication model by using the $ST_{HC_n}$.

With one port communication, the source node $s$ sends out packet (or element) $i$ for $0 \leq i < M$, to its subtrees through the edges of dimensions $n_t, 0, 1, \ldots, t - 1, t, n_{t+1}, t + 1, n_{t+2}, \ldots, n - 1, n_n$ during a time period $t_p$, where $t_p = \text{DEG}(s; 2^n)$.
The total broadcasting time is then $T = \text{DEG}(s; 2^n) \times M$. Since the height of the $ST_{HC_n}$ is $\lceil \frac{3}{2} \rceil$, the broadcasting time for all-port communication is $T = M + \lceil \frac{3}{2} \rceil - 1$.

7.2. Spanning Trees in $HC(N)$, where $2^{k-1} < N < 2^k$ and $k > 1$

With the help of the structures of spanning trees $ST_{HC_n}$, (incomplete) $BST$, and $B^+ST$, the construction of spanning trees in network $HC(N)$, denoted as $ST_{HC(N)}$, where $2^{k-1} < N < 2^k$ and $k > 1$ becomes simple. Let the decomposition be $HC(N) = \{HC_{p_1}, Q_{p_{i-1}}, \ldots, Q_{p_1}\}$. The spanning tree $ST_{HC(N)}$ rooted at an arbitrary source $s$ is constructed depending on the membership of node $s$ to the induced subgraphs: $s \in HC_{p_i}$ and $s \notin HC_{p_i}$.

Consider the case that node $s$ is in the subgraph $HC_{p_i}$. We can construct the $ST_{HC_{p_i}}$ rooted at $s$ by Algorithm $ST_{HC_n}$ such that the height of the $ST_{HC_{p_i}}$ is $\lceil \frac{p_i}{2} \rceil = \lceil \frac{k-1}{2} \rceil = \lceil \frac{\log N}{2} \rceil$. Using the $p_i$-dimensional $E1$-edge or $n_{p_{i+1}}$-dimensional $E2$-edge, a node $v_i$ in the subgraphs \{${Q_{p_{i-1}}, Q_{p_{i-2}}, \ldots, Q_{p_1}}$\} can be connected to the spanning tree $ST_{HC_{p_i}}$. If the Hamming distance $\rho(v_i, s) \leq h'$, then node $v_i$ is made the child of node $v_{i(p_{i+1})}$; else it is made the child of $v_{i(p_{i+1})}$, where $h' = (\frac{k}{2} + 1)$ when $k$ is even and $h' = \lceil \frac{3}{2} \rceil$ otherwise. In this case, the height of $ST_{HC(N)}$ is at most $\lceil \frac{\log N}{2} \rceil + 1$.

Consider the other case when node $s$ is not in the subgraph $HC_{p_i}$. Assume that $s$ is in $Q_{p_i}$ for $1 \leq i < l$. We can construct the incomplete binomial spanning tree $BST_{i+1}$ rooted at $s$ for subgraph $Q_{p_i}$ and the incomplete binomial trees $BST_{j}$ rooted at node $s_{[p_j+1]}$ for subgraphs $Q_{p_j}$, where $i < j < l$ and $\gamma = \lceil \frac{k-1}{2} \rceil + 1$. Then we append the incomplete binomial spanning trees $BST_{i}$ to $BST_{i+1}$ by making the roots $s_{[p_j+1]}$ as the children of $s$ through the $p_j$-dimensional $E1$-edges.

We can construct the $B^+ST$ rooted at $s_{[p_l+1]}$ for subgraph $HC_{p_l}$. Note that the
B^+ST is in turn composed of two incomplete binomial trees, $BST^p_\alpha$ rooted at node $r_1 = s^{p+i}$ and $BST^p_\beta$ rooted at node $r_2 = s^{(p+1)}$, where $\alpha = \beta = \left\lceil \frac{p}{2} \right\rceil$ if $p_i$ is odd; otherwise $\alpha - 1 = \beta = \left\lfloor \frac{p}{2} \right\rfloor$. We also append these two incomplete binomial trees $BST^p_\alpha$ and $BST^p_\beta$ to the $BT^p_{\gamma+1}$ in $Q_p$ by making the roots $r_1$ and $r_2$ as the children of $s$ through a $p_i$-dimensional $E$-edge and an $n_{(p_i+1)}$-dimensional $E$-edge, respectively.

The nodes in network $HC(N)$ which have not yet been included in the preceding incomplete binomial spanning trees are considered now. For the nodes $v_j$ in the subgraph $Q_pj$ for $1 \leq j \leq i - 1$, we compute the Hamming distance $h = \rho(v_j, s)$. If $h \leq \gamma$, where $\gamma = \left\lceil \frac{p_i}{2} \right\rceil + 1$ (the height of the incomplete binomial tree $BST^p_{\gamma+1}$ for $Q_p$), then make nodes $v_j$ as the children of $v_j^{(p_i+1)}$ in the $BST^p_{\gamma+1}$ through $p_i$-dimensional $E$-edges; else make them as the children of $v_j^{(p_i+1)}$ in the $BST^p_{\beta}$ (as mentioned above) through $n_{p_i+1}$-dimensional $E$-edges. For those nodes which are not used in the incomplete binomial spanning trees $BST^p_\gamma$ for subgraphs $Q_p$, we also connect them to the $BST^p_\beta$ through $n_{p_i+1}$-dimensional $E$-edges. The height of the spanning tree, $ST_{HC(N)}$, is bounded by $\left\lceil \frac{\log N}{2} \right\rceil + 1$.

Figures 7.3(a) and (b) show the spanning tree rooted at node $v_{19}$ in $HC(22) = \{HC_4, Q_2, Q_1\}$ and $B^+ST^5$ rooted at node $v_3$. Since $v_{19}$ is not in $HC_4$, by Algorithm $ST_{HC(N)}$, $B^+ST^5$ is spilt into two incomplete binomial trees rooted at nodes $v_3$ and $v_{12}$, respectively, then appended to the source node $v_6$ through the 4-dimensional $E$-edge and $n_5$-dimensional $E$-edge, respectively, as shown in Figure 7.3(a).

**Lemma 7.3.** The constructed $ST_{HC(N)}$ is the spanning tree of $HC(N)$ and its height is at most $\left\lceil \frac{\log N}{2} \right\rceil + 1$ for $N > 1$.

The detailed algorithm for the construction of the $ST_{HC(N)}$ in $HC(N)$ is presented below.
Figure 7.3: (a) The $ST_{HC}(2^2)$ rooted at node $v_{19}$. (b) The $B^+ST$ rooted at node $v_3$.

Algorithm $ST_{HC}(N)$: Construct the $ST_{HC}(N)$ in $HC(N)$, where $2^{k-1} < N < 2^k$.

1. Let $HC(N) = \{HC_{P_0}, Q_{P_1}, \ldots, Q_{P_l}\}$ where $N = \sum_{i=1}^{l} 2^{P_i}$, $1 \leq l \leq k$ and $P_{i+1} > P_i$.

2. if (the source node $s_i$ is in $HC_{P_i}$) then
   begin
   2.1 By Algorithm $ST_{HC_{P_i}}$, construct the $ST_{HC_{P_i}}$ rooted at $s$ for $HC_{P_i}$.
   2.2 Let $U_{P_i}^{P_i-1} = V_{P_i} \cup V_{P_{i+1}} \ldots \cup V_{P_{i-1}}$.
   2.3 if (k is even) then $h' = (\frac{k}{2} + 1)$ else $h' = \lceil \frac{k}{2} \rceil$.
   2.4 for each node $v_i \in U_{P_i}^{P_i-1}$ do
       begin
       2.5 Compute $\rho(v_i, s) = h$.
       2.6 if ($h \leq h'$) then make $v_i$ the child of $v_{i[P_i]}$ in $ST_{HC_{P_i}}$.
       2.7 else make $v_i$ the child of $v_{i(P_i+1)}$ in $ST_{HC_{P_i}}$.
       end
   end
   end
else begin \{ \text{s is not in } HC_p \} \\
3.1 Assume that \( s \in Q_{pi} \).
3.2 Let \( \gamma = \lceil \frac{k-1}{2} \rceil + 1 \).
3.3 Construct the \( BST_{i+1}^{pi} \) of \( Q_{pi} \) rooted at \( s \).
3.4 for each subgraph \( Q_{pi} \), where \( i < j < l \), do \\
begin\\
3.5 Construct the \( BST_{j}^{pi} \) rooted at \( s^{[p]} \).
3.6 Append the \( BST_{j}^{pi} \) to \( BST_{i+1}^{pi} \) by adding the edges \( < s, s^{[p]} > \).
end\\
3.7 Construct the \( B^+ST \) of \( HC_p \) rooted at \( s^{[p]} \).
3.8 Split the \( B^+ST \) into two subtrees \( BST_{a}^{pi} \) and \( BST_{b}^{pi} \).
3.9 Append \( BST_{a}^{pi} \) and \( BST_{b}^{pi} \) to \( BST_{i+1}^{pi} \) by adding new edges \( < s, s^{[p]} > \) and \( < s, s^{[p+1]} > \).
3.10 Let \( U_{pi-1}^{p} = V^{pi} \cup V^{p2} \ldots \cup V^{p_{i-1}} \).
3.11 for each \( v_j \in U_{pi-1}^{p} \) do \\
begin\\
3.12 Compute \( \rho(v_j, s) = h \).
3.13 if \( (h \leq \gamma) \) then make \( v_j \) the child of \( v_{j^{pi}} \) in \( BST_{i+1}^{pi} \).
3.14 else make \( v_j \) the child of \( v_{j^{[p+1]}} \) in \( BST_{b}^{pi} \).
end\\
3.15 Let \( U' \) be the set of nodes which are not used in \( BST_{i+1}^{pi} \) and \( BST_{j}^{pi} \) for \( i < j < l \).
3.16 for each \( v_j \in U' \) do \\
make \( v_j \) the child of \( v_{j^{[p+1]}} \) in \( BST_{b}^{pi} \).
end
We also consider two cases for the one-port communication in $ST_{HC}$ — the source node $s \in HC_{pi}$ or $s_i \in Q_{pi}$ for $1 \leq i < l$. For the first case, the broadcasting scheme is similar to that of spanning tree $ST_{HC_n}$. The source node $s$ sends out element $i$ for $0 \leq i < M$ to its children through the edges of dimensions $n_t, 0, 1, \ldots , t, n_{t+1}, \ldots , p_t-1, n_{p_t}, p_t$ (if it exists) during a time period $t_p = DEG(s; 2^k)$.

For the second case, $s$ sends out element $i$ to its children through the edges of dimensions $n_{p_t}, p_t - 1, n_{p_t-1}, p_t - 1, \ldots , n_{p_{t+1}}, p_{t+1} - 1, 0, 1, \ldots , p_t - 1$ also during a time period $DEG(s; 2^k)$. With one-port communication, the broadcasting time is $T = DEG(s; 2^k) \times M$. For all-port broadcast, $T = M + \lfloor \log_2 N \rfloor$.

7.3. Construction of Multiple Spanning Trees (MUST's)

In this section, we will construct another type of broadcasting trees, called multiple spanning trees (MUST's), in Hamming cubes. Note that an edge-disjoint spanning tree (EDST) is a directed spanning tree in which the edges in the tree are distinct. A directed MUST is composed of at least one EDST by making the last child of each root of EDST's as the common new root and reversing the edges from the roots of EDST's to the new root. The EDST's in a MUST are all edge-disjoint (in the sense of directed-edges). Let us first consider the construction of the MUST's in the $n$-dimensional Hamming cube, $HC_n$.

7.3.1. MUST's in $HC_n$

Taking advantage of the additional complementary edges in the folded hypercube $FQ_n$ [36], its embedded MUST consisting of $n + 1$ number of EDST's can be constructed. The height of the MUST in $FQ_n$ is $\lfloor \frac{n}{2} \rfloor + 1$, which is equal to the diameter plus one. Based on the fact that the folded hypercube $FQ_n$ is a spanning subgraph
of the $n$-dimensional Hamming cube $HC_n$, the embedded $MUST$ in $FQ_n$ also exists in $HC_n$, indicated as $MUST_{HC_n}$. Figure 7.4 shows the $MUST_{HC_4}$. Using the $MUST_{HC_n}$, the broadcasting time for one-port communication is $T = M + n - 1 + \epsilon$, where $\epsilon = 1$ if $n$ is even; otherwise $\epsilon = 0$. With all-port communication, $T = \lceil \frac{M}{n+1} \rceil + \lceil \frac{n}{2} \rceil$.

7.3.2. $MUST$'s in $HC(N)$, where $2^{k-1} < N < 2^k$ and $k > 1$

In the following, we construct the $MUST$'s for Hamming cubes whose size is not a power of two. This is denoted as $MUST_{HC(N)}$, where $2^k < N < 2^k$ and $k > 1$. In Section 6.5, we have seen how to construct the $MUST$'s in the generalized incomplete hypercubes. The technique can be modified to construct $MUST_{HC(N)}$. Importantly, the modification has to deal with the difference resulting from the network topologies, and also optimally utilizing the additional edges (namely, $E2$-edges) in the Hamming
cubes.

Since the generalized incomplete hypercubes $GIQ(N)$ of size $N$ for $2^{k-1} < N < 2^k$ and $k > 1$, are irregular and asymmetric, the construction of the $MUST_{GIQ(N)}$ has the following steps:

1. Decompose network $GIQ(N)$ into several subcubes according to the size $N$, i.e. $GIQ(N) = \{Q_{p_1}, Q_{p_1-1}, \ldots, Q_{p_1}\}$.

2. Assume that the source node $s_i$ is in subcube $Q_{p_i}$. According to the address of $s_i$ and the mapping function given in Definition 5.1, we determine the virtual source nodes $s'_j$ in subcubes $Q_{p_j}$, where $1 \leq j \leq l$ and $j \neq i$.

3. Rooted at the (virtual) source nodes, construct the embedded $MUST$ for each subcube $Q_{p_i}$ for $1 \leq i \leq l$. This $MUST$ consists of $p_i$ number of $EDST$'s.

4. Unite the embedded $MUST$'s in the subcubes to construct the $MUST$ in $GIQ(N)$.

Algorithm $MUST_{GIQ}$ which constructs the $MUST$ in $GIQ(N)$ can be applied to construct the $MUST$ in Hamming cubes $HC(N)$. However, the following modifications are necessary:

1. Since Hamming cubes are recursive networks, the decomposition is $HC(N) = \{HC_{p_1}, Q_{p_1-1}, \ldots, Q_{p_1}\}$. Compared to the decomposition of $GIQ(N)$, the subgraph of $HC_{p_i}$ is different. Thus, we construct the $MUST_{HC_{p_i}}$, rooted at the (virtual) source node $s'_{p_i}$ for the difference. Due to this, $p_i + 1$ number of $EDST$'s in $HC_{p_i}$ can be obtained.

2. Using the additional $E2$-edges of Hamming cubes, one more $EDST$ can be constructed in $MUST_{HC(N)}$ than in the $MUST_{GIQ(N)}$. Algorithm $MUST_{HC(N)}$
shows the construction of the MUST in HC(N). Note that Lines 7.3–7.4 and 8.2 in the algorithm create the extra EDST.

Algorithm MUST.HC(N): Construct the MUST of HC(N), MUST_{HC(N)}.
begin
1 Let \( N = \sum_{i=1}^{l} 2^{p_{i}} \), where \( 1 \leq l \leq k \) and \( p_{i+1} > p_{i} \).
2 Decompose \( HC(N) = \{HC_{p_{1}}, Q_{p_{1}-1}, \ldots, Q_{p_{1}}\} \) for which the \( l \) number of node-subsets are \( V^{p_{i}} = \{v_{\alpha} \mid 0 \leq \alpha < 2^{p_{i}}\} \) and \( V^{p_{i-1}} = \{v_{\alpha} \mid \sum_{j=i+1}^{l} 2^{p_{j}} \leq \alpha < \sum_{j=i+1}^{l} 2^{p_{j}}\} \) for \( 1 \leq i \leq l - 1 \).
3 Assume that the source node \( s_{i} \in V^{p_{i}} \).
4 By Definition 5.1, compute the virtual source nodes in each \( V^{p_{j}} \), where \( 1 \leq j \leq l \) and \( i \neq j \).
5 Construct the MUST for each subgraph \( \{HC_{p_{1}}, Q_{p_{1}-1}, \ldots, Q_{p_{1}}\} \) rooted at its corresponding (virtual) source node .
6 Let \( U_{p_{1}}^{p_{i}-1} = V^{p_{1}} \cup V^{p_{2}} \ldots \cup V^{p_{i-1}} \).
7 If \( (i \neq l) \) then \( \{s_{i} \text{ is not in } HC_{p_{i}}\} \)
    begin
    7.1 for \( j = 1 \) to \( i - 1 \) do
        call UNION.MUST.PHASE.1(\( MUST^{p_{j}}_{p_{i}}, MUST^{p_{j+1}}_{p_{i+1}}, MUST^{p_{j+1}}_{p_{i+1}} \)).
    7.2 for \( j = i \) to \( l - 1 \) do
        call UNION.MUST.PHASE.2(\( MUST^{p_{j}}_{p_{i}}, MUST^{p_{j+1}}_{p_{i+1}}, MUST^{p_{j+1}}_{p_{i+1}} \)).
    7.3 Let the \( (p_{i} + 1)\)th EDST of \( MUST^{p_{i}}_{p_{i}} \) rooted at \( s'_{p_{i}} \) be \( E'_{(p_{i}+1)} \) rooted at \( r'_{(p_{i}+1)} \).
    7.4 Create another EDST of \( MUST^{p_{i}}_{p_{i}} \) by linking \( s_{p_{i}} \) to \( r'_{(p_{i}+1)} \) through the \( n_{(p_{i}+1)}\)-dimensional edge and making each \( v_{t} \in (U_{p_{i}}^{p_{i-1}} - s_{p_{i}}) \) the child of \( v_{t_{(p_{i}+1)}} \) in \( E'_{(p_{i}+1)} \).
8 else begin \( \{s_{i} \text{ is in } HC_{p_{i}}\} \)
8.1 for $j = 1$ to $l - 1$ do

    call UNION\_MUST\_PHASE\_1($MUST_{P_1}^{P_1}, MUST_{P_1+1}^{P_1+1}, MUST_{P_1}^{P_1+1}$).

8.2 Create another EDST of $MUST_{P_1}^{P_1}$ by making each $v_i \in U_{P_1}^{P_1-1}$ the child of $v_{i(P_1+2)}$ in $E'_{(P_1+1)}$.

end

end.

Figure 7.5: The $MUST_{HC(27)}$ rooted at node $v_{22}$.

In Algorithm MUST\_HC(N), we take advantage of $E2$-edges to construct one more EDST in MUST\_HC(N) than those in MUST\_GQ(N). When the source node $s_i$ is not in HC$_{P_1}$, we link the $(P_1 + 1)$th EDST, $E_{(P_1+1)}$, of MUST$_{P_1}^{P_1}$ and link the nodes $v_i$ in subgraphs $\left\{Q_{P_1}, Q_{P_1-1}, \ldots, Q_{P_1}\right\}$ (i.e. $v_i \in U_{P_1}^{P_1-1}$) to their corresponding nodes in $E_{(P_1+1)}$ through the $n_{(P_1+1)}$-dimensional edges. When $s_i \in HC(2^{P_1})$, we simply link the node $v_i \in U_{P_1}^{P_1-1}$ to $v_{i(P_1+1)}$ through the $n_{(P_1+1)}$-dimensional edges. It can be seen that the edges to construct MUST$_{P_1}^{P_1}$ by Algorithm MUST\_HC(N) are edge-disjoint. Therefore, the following theorem is immediately obtained.
Theorem 7.1. The \( \text{MUST}_H(N) \), where \( 2^{k-1} < N < 2^k \), is composed of \( \phi_{\min}(H(N)) \) edge-disjoint spanning trees and has the height of \( \left \lfloor \frac{\log N}{2} \right \rfloor + 2 \leq h \leq 3[\log N] - 2 \).

Figure 7.5 shows the \( \text{MUST}_H(27) \) rooted at node \( v_{22} \). The five triangular shapes denote the five \( EDST \)'s in \( \text{MUST}_H^4 \) rooted at node \( v_6 \). The fifth \( EDST \) is appended to the source node through the \( n_5 \)-dimensional \( E2 \)-edge. Using \( \text{MUST}_H(N) \), the broadcast time for one-port communication is \( M + h - 1 \) and for all-port communication is \( \left \lfloor \frac{M}{\phi_{\min}(GQ(N)) + 1} \right \rfloor + h \), where \( \left \lfloor \frac{\log N}{2} \right \rfloor + 2 \leq h \leq 3[\log N] - 2 \).

7.4. Summary

Table 7.1: Time complexity of communication.

<table>
<thead>
<tr>
<th>Broadcast trees</th>
<th>one-port</th>
<th>all-port</th>
</tr>
</thead>
<tbody>
<tr>
<td>( ST_{Q_4} )</td>
<td>( n \times M )</td>
<td>( M + n - 1 )</td>
</tr>
<tr>
<td>( ST_{THC} )</td>
<td>( DEG(s;2^n) \times M )</td>
<td>( M + \left \lfloor \frac{n}{2} \right \rfloor - 1 )</td>
</tr>
<tr>
<td>( ST_{GQ(N)} )</td>
<td>( \lceil \log N \rceil \times M )</td>
<td>( M + \left \lfloor \frac{\log N}{2} \right \rfloor - 1 )</td>
</tr>
<tr>
<td>( ST_{HC(N)} )</td>
<td>( DEG(s;2^k) \times M )</td>
<td>( M + \left \lfloor \frac{\log N}{2} \right \rfloor )</td>
</tr>
<tr>
<td>( ST_{BGIQ(N)} )</td>
<td>( \left \lfloor \log N \right \rfloor + 1 \times M )</td>
<td>( M + \left \lfloor \frac{\log N}{2} \right \rfloor )</td>
</tr>
<tr>
<td>( MUST_{Q_4} )</td>
<td>( M + n - 1 )</td>
<td>( \left \lfloor \frac{M}{n+1} \right \rfloor + \left \lfloor \frac{n}{2} \right \rfloor - 1 )</td>
</tr>
<tr>
<td>( MUST_{THC} )</td>
<td>( M + n - 1 + \varepsilon )</td>
<td>( \left \lfloor \frac{M}{n+1} \right \rfloor + \left \lfloor \frac{n}{2} \right \rfloor )</td>
</tr>
<tr>
<td>( MUST_{GQ(N)} )</td>
<td>( M + h_1 - 1 )</td>
<td>( \left \lfloor \frac{M}{\phi_{\min}(GQ(N))} \right \rfloor + h_1 - 1 )</td>
</tr>
<tr>
<td>( MUST_{HC(N)} )</td>
<td>( M + h_2 - 1 )</td>
<td>( \left \lfloor \frac{M}{\phi_{\min}(GQ(N)) + 1} \right \rfloor + h_2 )</td>
</tr>
<tr>
<td>( MUST_{BGIQ(N)} )</td>
<td>( M + h_3 - 1 )</td>
<td>( \left \lfloor \frac{M}{\phi_{\min}(GQ(N))} \right \rfloor + h_3 )</td>
</tr>
</tbody>
</table>

\( \left \lfloor \frac{\log N}{2} \right \rfloor + 1 \leq h_1 \leq 3[\log N] - 2 \), \( \left \lfloor \frac{\log N}{2} \right \rfloor + 2 \leq h_2 \leq 3[\log N] - 2 \).

\( \left \lfloor \frac{\log N}{2} \right \rfloor + 3 \leq h_3 \leq 3[\log N] - 1 \).

\( DEG(\alpha;\beta) = \) degree of node \( v_\alpha \) in \( HC(\beta) \), where \( 0 \leq \alpha < \beta \).

\( M \) is the number of broadcasted elements.

In this chapter, we have constructed two embedded broadcasting trees, the span-
ning trees and multiple spanning trees, in Hamming cube $HC(N)$. For the case when $N = 2^n$, the spanning tree $ST_{HC_n}$ has the height equal to its diameter $\lceil \frac{n}{2} \rceil$ which is half of that in the binary hypercube $Q_n$. The $MUST_{HC_n}$ consists of $(n + 1)$ EDST's and has the height of $\lceil \frac{n}{2} \rceil + 1$. Also the $MUST_{HC_n}$ has one more EDST than in $Q_n$.

For the case when $2^{k-1} < N < 2^k$, the height of $ST_{HC(N)}$ is reduced to $\lceil \frac{\log N}{2} \rceil + 1$, almost half of the height of $ST_{GIQ(N)}$, by using the enhanced $E2$-edges of $HC(N)$. Also by the $E2$-edges in $HC(N)$, one more EDST can be constructed in the $MUST_{HC(N)}$ of height $h_2$, where $\lceil \frac{\log N}{2} \rceil + 2 \leq h_2 \leq 3\lceil \log N \rceil - 2$.

Table 7.1 shows the time complexity for one-port and all-port communication models by using the embedded broadcasting trees in generalised incomplete hypercubes, Hamming cubes, enhanced generalized incomplete hypercubes (introduced in the next chapter).
CHAPTER 8

BROADCASTING TREES IN ENHANCED GENERALIZED INCOMPLETE HYPERCUBES

In this chapter, we design a broadcasting scheme for the enhanced generalized incomplete hypercubes with the help of the embedded broadcasting trees, the spanning trees and multiple spanning trees.

In Chapter 6 and 7, we have introduced the binomial spanning trees ($BST$) which are the embedded broadcasting trees in the binary hypercubes, and the binomial plus spanning trees ($B^+ST$) which are used as a building block for constructing the spanning trees in Hamming cubes. There exists another variant of $BST$, called spanning broadcasting trees ($SBT$), which has been constructed for broadcasting in the folded hypercubes [36]. With the help of these trees and the concept of link replacement (introduced in Chapter 5), we construct the embedded spanning trees in the enhanced generalized incomplete hypercubes.

In Chapter 7, we have seen how to apply the algorithm for constructing the multiple spanning trees in generalized incomplete hypercubes to the case of Hamming cubes. This technique can also be used here to construct the multiple spanning trees in the enhanced generalized incomplete hypercubes $EGIQ(N)$, where $2^n < N < 2^{n+1}$ and $n > 1$.

8.1. Construction of Spanning Trees, $ST_{EGIQ(N)}$'s

In this section, we provide an algorithm to construct the embedded spanning trees $ST_{EGIQ(N)}$ in the network $EGIQ(N)$, where $2^n < N < 2^{n+1}$. The tree con-
Construction is based on the concept of link replacement and three types of spanning trees, namely binomial spanning trees BST, binomial plus spanning trees $B^+ST$, and spanning broadcasting trees SBT.

The embedded spanning broadcasting trees SBT in the folded hypercubes [36] are defined by a parent-to-children relation which is similar to that of BST. Taking advantage of the complementary edges in the folded hypercubes, the embedded spanning broadcasting tree has the height of $\lceil \frac{n}{2} \rceil$, equal to the diameter of the folded hypercubes. Interestingly, the number of nodes at different levels of the tree can also be characterized by the formula in Property 7.1. Figure 8.1 shows the embedded SBT rooted at node $s$ in the folded hypercube $FQ_5$. In this figure, the numbers on the tree edges are their dimensions and the notation $n_5$ stands for a 5-dimensional complementary edge, which is distinguished from a regular hypercube edge. A spanning broadcasting tree rooted at node $v_0$ can work as a template from which the other spanning broadcasting trees can be translated by the bitwise Exclusive-OR operation.

By Section 4.3, decomposition $EGIQ(N) = \{X_{p_1}, Q_{p_1-1}, \ldots, Q_{p_1}\}$, where $N =$
The subgraph $X_{p_i} = IFQ_{p_i}$, the incomplete folded hypercube (Section 5.3), for $2^n < N \leq 2^n + 2^{n-1}$ and $X_{p_i} = Q_{p_i}$ for $2^n + 2^{n-1} < N < 2^{n+1}$. Let us consider two cases depending on if the source node $s$ is in the subgraph $X_{p_i}$ for the construction of spanning trees $ST_{EGIQ(N)}$ in $EGIQ(N)$. Based on the concept of link replacement and the structures of $B^+ ST$, $BST$, and $SBT$, the construction of the spanning tree in $EGIQ(N)$ is given by the following algorithm.

**Algorithm ST. EGIQ(N):** Construct the spanning trees in $EGIQ(N)$.

begin
1. Let $EGIQ(N) = \{X_{p_1}, Q_{p_{l-1}}, \ldots, Q_{p_1}\}$, where $N = \Sigma_{i=1}^{l} 2^{p_i}$, $1 \leq l \leq n + 1$, and $p_{i+1} > p_i$. Subgraph $X_{p_i} = IFQ_{p_i}$ for $2^n < N \leq 2^n + 2^{n-1}$ and $X_{p_i} = Q_{p_i}$ for $2^n + 2^{n-1} < N < 2^{n+1}$.
2. if (the source node $s$ is in $X_{p_i}$) then
   begin
   2.1. Construct the spanning broadcasting tree $SBT$ rooted at $s$ for $X_{p_i}$.
   2.2. Let $U_{p_i}^{p_i-1} = V_{p_1} \cup V_{p_2} \ldots \cup V_{p_{i-1}}$.
   2.3. if ($n + 1$ is even) then $h' = \lceil \frac{n+1}{2} \rceil$ else $h' = \lfloor \frac{n+1}{2} \rfloor$.
   2.4. for each node $v_t \in U_{p_i}^{p_i-1}$ do
      begin
      2.5. Compute $\rho(v_t, s) = h$.
      2.6. if ($h > h'$) then Make $v_t$ the child of $v_{t[p_{i+1}]}$ in $SBT$.
      end
      else begin
      2.7.1. Make $v_t$ the child of $v_{\alpha}$, where $v_{\alpha} = v_{[p_{i+1}]}$ in $SBT$.
      2.7.2. if ($v_{\alpha}$ has a child $v_{\beta}$ such that $v_{\beta} = v_{[p_{i+1}]}$) then
      2.7.3. Make $v_{\beta}$ the child of $v_t$ through a $n_{p_{i+1}}$-dimensional link.
      end
end
end

end

3 $\text{else begin \{s is not in } X_p_i \}$

3.1 Assume that $s \in Q_p_i$.

3.2 Let $\gamma = \lceil \frac{n}{2} \rceil + 1$.

3.3 Construct the $BST^p_{\gamma+1}$ rooted at $s$ for $Q_p_i$.

3.4 for each $Q_{p_j}$, where $i < j \leq l - 1$ do

begin

3.5 Construct the $BST^p_j$ rooted at $s^{[p_j+1]}$.

3.6 Append the $BST^p_j$ to the $BST^p_{\gamma+1}$ by adding the edges $< s, s^{[p_j]} >$.

end

3.7 Construct the $B^+ST$ rooted at $s^{[p_l+1]}$ for $X_p_i$.

3.8 Split the $B^+ST$ into two composite subtrees $BST^p_\alpha$ and $BST^p_\beta$.

3.9 Append $BST^p_\alpha$ and $BST^p_\beta$ to the $BST^p_{\gamma+1}$ of $Q_p_i$ by adding new edges $< s, s^{[p_l+1]} >$ and $< s, s^{[p_l+1]} >$.

3.10 Let $U_{p_l}^{p_{l-1}} = V_{p_l} \cup V_{p_{l-1}}$.

3.11 for each $v_t \in U_{p_l}^{p_{l-1}}$ do

begin

3.12 Compute $\rho(v_t, s) = h$.

3.13 if $(h \leq \gamma)$ then make $v_t$ the child of $v_{i(p_l+1)}$ in the $BST^p_{\gamma+1}$ of $Q_p_i$.

3.14 else make $v_t$ the child of $v_{i(p_l+1)}$ in the $BST^p_\beta$.

end

3.15 Let $U'$ be the set of nodes which are not used in the $BST^p_{\gamma+1}$ and $BST^p_\gamma$,

where $i < j < l$.

3.16 for each $v_t \in U'$ do
make \(v_t\) the child of \(v_{t(p_t+1)}\) in the \(BST_{p_t}^{\beta}\).

end

end

Algorithm \(ST\_EGIQ(N)\) considers two cases to construct the spanning tree of \(EGIQ(N)\), depending on whether the source node \(s\) is in \(V^{p_t}\) (with respect to subgraph \(X_{p_t}\)). If \(s\) is in \(X_{p_t}\), then we first construct the \(SBT\) rooted at \(s\) in \(X_{p_t}\). Note that the height of the \(SBT\) in \(X_{p_t}\) is given by \(\lceil \frac{\log n}{2} \rceil = \lceil \frac{n}{2} \rceil = \lceil \frac{\log N}{2} \rceil\). The nodes \(v_t\) in the subgraphs \(\{Q_{p_t-1}, Q_{p_t-2}, \ldots, Q_{p_t}\}\) are then included in the \(SBT\) as the children of \(v_{t(p_t+1)}\) (resp. \(v_{t(p_t+1)}\)) if \(\rho(v_t, s) = h > h'\) (resp. \(h \leq h'\), where \(h'\) is computed in Line 2.3. Lines 2.7.2—2.7.3 deal with the occurrence of link replacement. It can be seen that the height of the spanning tree constructed in this case is at most \(\lceil \frac{\log N}{2} \rceil + 1\).

Figure 8.2 gives an example for this case. The tree indicated by solid lines is the \(SBT\) rooted at node \(v_1\) in the folded hypercube \(FQ_3\). The broken lines are those dimensional links to include nodes \(v_t\) for \(8 \leq t \leq 15\). Note that when node \(v_9\) is included, the \(n_3\)-dimensional \(C3\)-link which connects nodes \(v_6\) and \(v_1\) will be replaced by the \(n_4\)-dimensional \(C2\)-link. Therefore, node \(v_6\) is made as the child of \(v_9\).
Consider the second case that node \( s \) is not in subgraph \( X_p \). Assume that \( s \in Q_p \) for \( i \neq l \). For subgraph \( Q_p \), we construct the incomplete binomial spanning trees \( BST_{\gamma+1}^p \) rooted at \( s \), where \( \gamma = \lceil \frac{n}{2} \rceil + 1 \) is given by Line 3.2. Then for each subgraph \( Q_p \), where \( i < j < l \), we construct the incomplete binomial tree \( BST_{\gamma}^p \) rooted at node \( s^{[p_i+1]} \). The incomplete binomial spanning trees \( BST_{\gamma}^p \) are then appended to \( BST_{\gamma+1}^p \) through a \((p_j)\)-dimensional \( C1 \)-link.

We can construct the \( B^+ST \) rooted at \( s^{[p_i+1]} \) for the subgraph \( X_p \). The \( B^+ST \) can be split into two incomplete binomial trees according to the discussion of Section 6.1.1.2., and then appended to the \( BST_{\gamma+1}^p \) of \( Q_p \).

Lines 3.10—3.14 include the nodes in subgraphs \( \{Q_{p_i-1}, Q_{p_i-2}, \ldots, Q_{p_i}\} \) are included in the \( BST_{\gamma+1}^p \) of \( Q_p \) or the \( BST_{\gamma}^p \) according to their Hamming distance with the source node. Lines 3.14—3.16 include the remaining nodes that have not yet been considered.

Note that in this case, we construct a \( B^+ST \) for subgraph \( X_p \), then split it into two incomplete binomial trees. The edges of these two trees consist of only the underlying hypercube edges (corresponding to the \( E1 \)-edges of Hamming cubes). Therefore, the spanning trees \( ST_{EGIQ(N)} \) in this case are identical to the spanning trees \( ST_{HC(N)} \) when nodes \( s \notin HC_p \) as shown as in Section 6.1.2.

The height of this spanning tree is also bounded by \( \lceil \frac{\log N}{2} \rceil + 1 \). From the above analyses, we conclude

**Lemma 8.1.** Algorithm \( ST_{EGIQ(N)} \) constructs the spanning tree of \( EGIQ(N) \) of height at most \( \lceil \frac{\log N}{2} \rceil + 1 \) for \( N > 1 \).

Consider two cases for the one-port communication in \( EGIQ(N) \) by using the embedded spanning tree — the source node \( s \in X_p \) and \( s \in Q_p \) for \( i \neq l \). For the first case, the source node \( s \) sends out element \( i \) for \( 0 \leq i < M \) to its children
through the links of dimensions 0, 1, ..., \( p_i, (p_i + 1, n_{p_i+1} \text{ if exists}) \) during a time period \( t_p = \lceil \log N \rceil + 1 \). For the second case, \( s \) sends out element \( i \) to its children through the links of dimensions \( p_i + 1, n_{p_i+1}, 0, 1, ..., p_i \) during a time period \( t_p \). With one-port communication, the total broadcast time is \( T = (\lceil \log N \rceil + 1) \times M \). For all-port broadcast, \( T = M + \lceil \frac{\log N}{2} \rceil \).

### 8.2. Construction of Multiple Spanning Trees, MUST\( _{EGIQ(N)} \)'s

Now we modify the algorithm which constructs the multiple spanning trees in the generalized incomplete hypercubes to construct the multiple spanning trees MUST\( _{EGIQ(N)} \) in the enhanced generalized incomplete hypercubes \( EGIQ(N) \), where \( 2^n < N < 2^{n+1} \).

Due to the decomposition \( EGIQ(N) = \{X_{p_1}, Q_{p_1}, \ldots, Q_p\} \), the induced node subsets are given by \( V^p = \{v_\alpha \mid 0 \leq \alpha < 2^{p_i}\} \) and \( V^{p_l(p_1)} = \{v_\alpha \mid \sum_{j=1}^{l-i} 2^{p_j} \leq \alpha < \sum_{j=1}^{l-i} 2^{p_j}\} \) for \( 1 \leq i \leq l - 1 \), respectively. Assume that the source node \( s_i \) is in \( V^p \), i.e. in subgraph \( Q_{p_i} \) for \( 1 \leq i \leq l - 1 \) and is in \( X_{p_l} \) for \( i = l \). By Definition 5.1, the virtual source nodes \( s_j \) for \( 1 \leq j \leq l \) and \( j \neq i \), in the other subgraphs can be determined. Also, by Section 6.2.1, we can construct \( p_i \) edge-disjoint spanning trees (EDST’s) for each subgraph \( Q_{p_i} \), where \( 1 \leq i \leq l - 1 \).

Let us consider the decomposition of Hamming cubes, \( HC(N) = \{HC_{p_1}, Q_{p_1}, \ldots, Q_{p_1}\} \). In Section 6.2.2, we construct the MUST\( _{HC_{p_1}} \) for subgraph \( HC_{p_1} \), which is the MUST\( _Q \) and consists of \( n + 1 \) EDST’s. We know that due to the link-replacement, subgraph \( X_{p_l} = IFQ_{p_l} \) or \( Q_{p_l} \) depending on the network size. Thus, for network \( EGIQ \), we also construct the MUST\( _{HC_{p-l-1}} \) for subgraph \( X_{p_l} \) and then modify the tree by the link-replacement scheme. With this approach, Algorithm MUST\( _{HC(N)} \) can be reused. We merely need to add the following code at the end of the algorithm.
Figure 8.3: The MUST rooted at \( v_6 \) in \( EGIQ(11) \).

for each \( v_\alpha \in V^{p_l} \) do

begin

if a node \( v_\beta = v_\alpha^{p_l} \) has the child \( v_\gamma \) in the same \( EDST \) such that \( v_\gamma = v_\beta^{p_l} \), i.e.

they are connected through an \( n_{p_l} \)-dimensional enhanced \( C_2 \)-link.

then if \( v_\beta < v_\gamma \) then

make \( v_\gamma \) the child of \( v_\beta^{p_l+1} \) through a \( n_{p_l+1} \)-dimensional \( C_3 \)-link.

else make \( v_\gamma \) the child of \( v_\beta^{p_l+1} \) through a \( p_l \)-dimensional \( C_1 \)-link.

end

Figure 8.3 shows the MUST rooted at \( v_6 \) in \( EGIQ(11) \). Note that the bold broken lines indicate the link-replacement. For example, in the second \( EDST \) the \( n_3 \)-dimensional \( C_3 \)-links, \( < v_0, v_7 > \) is replaced by the \( n_4 \)-dimensional \( C_2 \)-links, such as say \( < v_8, v_7 > \). There are three, the minimum degree of \( EGIQ(11) \), \( EDST \)'s in the MUST. They are the second, third, and fourth \( EDST \)'s in the MUST.
General speaking, due to the link replacement, the last two EDST's in MUST could use the same links twice. More specifically, when the source nodes \( s_{p_i} \in X_{p_i} \) and \( s_{p_i-1} \in Q_{p_i-1} \) satisfy the relation \( s_{p_i} = s_{p_i+1}^{[p_i+1]} \) (or \( s_{p_i-1} = s_{p_i+1}^{[p_i+1]} \)), the \( p_i \)-dimensional C1-link \( < s_{p_i}, s_{p_i-1} > \) will be used twice in the \((p_i)\)th and \((p_i-1)\)th EDST's. In addition, it has been proven in Theorem 7.1 that Algorithm MUST\_HC(N) constructs the minimum degree \( \phi_{\min}(HC(N)) = \phi_{\min}(GIQ(N) + 1) \) number of EDST's in MUST\_HC(N) for the Hamming cube HC(N). Therefore, the following theorem is obtained.

**Theorem 8.1.** The MUST in EGIQ(N), where \( 2^{n-1} < N < 2^n \), is composed of \( \phi_{\min}(EGIQ(N)) - 1 = \phi_{\min}(GIQ(N)) \) edge-disjoint spanning trees and has the height of \( \left\lceil \frac{\log N}{2} \right\rceil + 3 \leq h \leq 3\left\lceil \log N \right\rceil - 1 \).

Using MUST\_EGIQ(N), the broadcast time for one-port communication is \( M + h - 1 \) and for all-port communication is \( \left\lceil \frac{M}{\phi_{\min}(GIQ(N))} \right\rceil + h \), where \( \left\lceil \frac{\log N}{2} \right\rceil + 3 \leq h \leq 3\left\lceil \log N \right\rceil - 1 \).

### 8.3. Summary

The binomial spanning trees (BST), the binomial plus spanning trees (B+ST), and spanning broadcasting trees (SBT) are embedding broadcasting trees in the binary hypercubes, Hamming cubes, and folded hypercubes, respectively. With these trees and the concept of link replacement, we have constructed the embedded spanning trees ST\_EGIQ(N) in enhanced generalized incomplete hypercubes. ST\_EGIQ(N) has height bounded by diameter \( \left\lceil \frac{\log N}{2} \right\rceil + 1 \) for \( N > 1 \). As in Section 6.2.2, we modified the algorithm which constructs the multiple spanning trees MUST\_GIQ(N) in generalized incomplete hypercubes to construct the multiple spanning trees MUST\_EGIQ(N) in enhanced generalized incomplete hypercubes. Because of the link replacement, the
height and the number of edge-disjoint spanning trees of $MUST_{EGIQ}(N)$ are almost the same as those of $MUST_{HC}(N)$, as shown in Table 7.1.
CHAPTER 9
PERFORMANCE ANALYSES

In this chapter, we evaluate the performance parameters such as average distance, network cost, and message traffic density of Hamming cubes and enhanced generalized incomplete hypercubes. We also compare them with those of binary hypercubes and incomplete hypercubes.

9.1. Average Distance

The diameter of a network provides the information of maximum number of message traversal between any two nodes. The worst case of message transmission latency is bounded by this structural property. For a typical message, the mean message traversal is indicated by the average distance of the network, which is defined as the average distance of all pairs of nodes. In the following, we first derive the average distance for the n-dimensional Hamming cube $HC_n$.

9.1.1. Average Distance of Network $HC_n$

In $HC_n$, we can embed a spanning tree ($ST_{HC_n}$) rooted at an arbitrary node (discussed in Chapter 7). The embedded spanning tree $ST_{HC_n}$, a variant of binomial spanning tree, is greedily constructed to span all nodes. Each path from the source node (the root of tree) to others has the shortest length. Figure 7.2 shows a spanning tree in network $HC_5$.

Consider the decomposition $HC_n = \{HC_2, Q_2, Q_3, \ldots, Q_{n-1}\}$. By Property 4.7, the nodes in the same induced subgraph have the same degree and satisfy the node-
symmetry. Thus, the embedded spanning trees $ST_{HC_n}$ rooted at the different source
nodes in the same subgraph are isomorphic. Therefore, the average distance of the
$n$-dimensional Hamming cube, $HC_n$, can be derived from the geometrical structure
of $ST_{HC_n}$.

The recursive decomposition yields: $HC_n = \{HC_2, Q_2, Q_3, \ldots, Q_{n-1}\} = \{HC_3, Q_3,
\ldots, Q_{n-1}\} = \{HC_4, Q_4, \ldots, Q_{n-1}\} = \ldots$. For convenience, in this section a node in
subgraph $Q_{i-1}$ is said to be in $HC_i$ for $3 \leq i \leq n - 1$. Let $m_{i,j}^k$ denote the number
of nodes at the $i$th level of $ST_{HC_j}$ (in $HC_j$) rooted at a source node in $HC_k$ (in the
subgraph $Q_{k-1}$ for $k \geq 3$), where $0 \leq i \leq k \leq j$. Then $m_{i,j}^k$ satisfies the following
relations.

**Property 9.1.** The number of nodes, $m_{i,j}^k$, at the $i$th level of $ST_{HC_j}$ rooted at a
source node in $HC_k$, where $0 \leq i \leq k \leq j$, is given as follows.

1. $m_{0,j}^k = 1$ for $2 \leq k \leq j$.
2. If $k$ is even then
   
   (i) $m_{i,k}^k = \begin{cases} \binom{\frac{k}{2}+1}{i} & \text{for } 1 \leq i \leq \frac{k}{2}, \\ 0 & \text{for } i > \frac{k}{2}. \end{cases}$

   (ii) $m_{1,k+1}^k = m_{1,k}^k + 2$ and $m_{2,k+1}^k = m_{2,k}^k + 2(m_{1,k}^k - \epsilon)$,
       where $\epsilon = 2$, if $k = 2$ and $\epsilon = 1$, otherwise.

   (iii) $m_{i,k+1}^k = m_{i,k}^k + 2m_{i-1,k}^k$ for $3 \leq i \leq \frac{k}{2}$; and
       $m_{\frac{k}{2}+1,k+1}^k = 2^{k+1} - \sum_{i=0}^{\frac{k}{2}} m_{i,k+1}^k$ and
       $m_{i,k+1}^k = 0$ for $i > \frac{k}{2} + 1$.

   (iv) when $j > k + 1$,
       $m_{i,j}^k = \begin{cases} m_{i,j-1}^k + 2m_{i-1,j-2}^k & \text{for } 1 \leq i \leq \left[\frac{j}{2}\right], \\ 0 & \text{for } i > \left[\frac{j}{2}\right]. \end{cases}$
(3) If \( k \) is odd then

\[
(i) \quad m_{i,k}^k = \begin{cases} \binom{k+1}{i} & \text{for } 1 \leq i \leq \lfloor \frac{k}{2} \rfloor, \\ \frac{1}{2} \binom{k+1}{i} & \text{for } i = \lfloor \frac{k}{2} \rfloor, \\ 0 & \text{for } i > \lfloor \frac{k}{2} \rfloor. \end{cases}
\]

\[
(ii) \quad m_{1,k+1}^k = m_{1,k}^k + 2 \quad \text{and} \quad m_{2,k+1}^k = m_{2,k}^k + 2(m_{1,k}^k - 1).
\]

\[
(iii) \quad m_{i,k+1}^k = m_{i,k}^k + 2m_{i-1,k}^k \quad \text{for } 3 \leq i \leq \lfloor \frac{k}{2} \rfloor - 1; \quad \text{and}
\]

\[
m_{\lfloor \frac{k}{2} \rfloor, k+1}^k = 2^{k+1} - \sum_{i=0}^{\lfloor \frac{k}{2} \rfloor - 1} m_{i,k+1}^k \quad \text{and}
\]

\[
m_{i,k+1}^k = 0 \quad \text{for } i > \lfloor \frac{k}{2} \rfloor.
\]

(iv) when \( j > k + 1, \)

\[
m_{i,j}^k = \begin{cases} m_{i,j-1}^k + 2m_{i-1,j-2}^k & \text{for } 1 \leq i \leq \lfloor \frac{j}{2} \rfloor, \\ 0 & \text{for } i > \lfloor \frac{j}{2} \rfloor. \end{cases}
\]

Table 9.1: \( m_{i,j}^2 \), where \( 0 \leq i \leq 5 \) and \( 2 \leq j \leq 10 \).

<table>
<thead>
<tr>
<th>( i \cap j )</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>5</td>
<td>7</td>
<td>9</td>
<td>11</td>
<td>13</td>
<td>15</td>
<td>17</td>
<td>19</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>8</td>
<td>18</td>
<td>32</td>
<td>50</td>
<td>72</td>
<td>98</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>20</td>
<td>56</td>
<td>120</td>
<td>220</td>
<td>364</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>48</td>
<td>160</td>
<td>400</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>16</td>
<td>112</td>
<td></td>
</tr>
</tbody>
</table>

Example 9.1: Table 9.1 and 9.2 show \( m_{i,j}^2 \), where \( 0 \leq i \leq 5 \) and \( 2 \leq j \leq 10 \), and \( m_{i,j}^4 \), where \( 0 \leq i \leq 5 \) and \( 4 \leq j \leq 10 \), respectively. Figure 7.2 shows \( ST_{HC_5} \) rooted at node \( v_{10} \). The subtree of \( ST_{HC_5} \) indicated by solid lines is the \( ST_{HC_4} \) rooted at \( v_{10} \). Note that node \( v_{10} \) belongs to \( HC_4 \). Therefore, by Table 9.2, the number of nodes at the \( i \)th level of \( ST_{HC_4} \) are 1, 5, and 10 corresponding to \( m_{i,4}^4 \), where \( 0 \leq i \leq 2 \). Also,
Table 9.2: $m_{i,j}^4$, where $0 \leq i \leq 5$ and $4 \leq j \leq 10$.

<table>
<thead>
<tr>
<th>$i \backslash j$</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>7</td>
<td>9</td>
<td>11</td>
<td>13</td>
<td>15</td>
<td>17</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>18</td>
<td>28</td>
<td>42</td>
<td>60</td>
<td>82</td>
<td>108</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td>26</td>
<td>62</td>
<td>118</td>
<td>202</td>
<td>322</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>12</td>
<td>64</td>
<td>188</td>
<td>424</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>24</td>
<td>152</td>
<td></td>
</tr>
</tbody>
</table>

the number of nodes at the $i$th level of $ST_{HC_n}$ are $m_{i,5}^4$, for $0 \leq i \leq 3$, whose values are 1, 7, 18, and 6.

With the help of tables of $m_{i,j}^k$'s and Property 4.7, the average distance $\bar{d}_{HC_n}$ of $HC_n$ is given by the following property.

**Property 9.2.** Let $N = 2^n$. Then the average distance of $HC_n$ is

$$\bar{d}_{HC_n} = \frac{1}{N-1} \left\{ \sum_{k=2}^{n} (2^{k-n-1} \sum_{i=1}^{\lfloor \frac{n}{k} \rfloor} i \cdot m_{i,n}^k) + \frac{2^{\lfloor \frac{n}{2} \rfloor}}{N} \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} i \cdot m_{i,n}^2 \right\}. \quad (9.1)$$

**Proof:**

$$\bar{d}_{HC_n} = \frac{1}{N(N-1)} \left\{ \sum_{k=3}^{n} (2^{k-n-1} \sum_{i=1}^{\lfloor \frac{n}{k} \rfloor} i \cdot m_{i,n}^k) + 4 \cdot \sum_{i=1}^{\lfloor \frac{n}{4} \rfloor} i \cdot m_{i,n}^4 \right\} \quad (9.2)$$

$$= \frac{1}{N(N-1)} \left\{ \sum_{k=2}^{n} (2^{k-n-1} \sum_{i=1}^{\lfloor \frac{n}{k} \rfloor} i \cdot m_{i,n}^k) + 2 \cdot \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} i \cdot m_{i,n}^2 \right\} \quad (9.3)$$

Note that in the proof of Property 9.2, we rewrite Equation (9.2) as Equation (9.3) to prevent the accumulation error when $n$ is very large. It is known that the average distance of the $n$-dimensional binary hypercube is $\bar{d}_{Q_n} = \frac{\sum_{i=1}^{n} \binom{n}{i}}{N-1} = \frac{nN}{2(N-1)}$. The
average distance of the folded hypercube [36] is given by $d_{FQ_n} = \frac{\sum_{i=1}^{\left\lfloor \frac{n}{2} \right\rfloor} ni_i}{N-1}$, where $n_i = \binom{t+1}{i}$ for $0 \leq i \leq \left\lfloor \frac{n}{2} \right\rfloor$ when $n$ is even; otherwise, $n_i = \begin{cases} \binom{t+1}{i} & \text{for } 0 \leq i \leq \left\lfloor \frac{n}{2} \right\rfloor, \\ \frac{1}{2} \binom{t+1}{i} & \text{for } i = \left\lceil \frac{n}{2} \right\rceil, \end{cases}$.

Figure 9.1 shows the average distance of networks $Q_n$, $FQ_n$, and $HC_n$ as a function of network dimensions. Due to the additional edges ($E_2$-edges) and recursive nature, the average distance of Hamming cube is less than those of hypercubes and folded hypercubes.

9.1.2. Average Distance of Networks $HC(N)$ and $EGIQ(N)$

With the help of an all-to-all shortest path algorithm, we have computed the average distances of networks $EGIQ(N)$, $HC(N)$, and compared to that of $GIQ(N)$,
Figure 9.2: The average distances $\bar{d}_{EGIQ(N)}$, $\bar{d}_{HC(N)}$, and $\bar{d}_{GIQ(N)}$.

where $2^n < N < 2^{n+1}$ and $n \geq 1$. Figure 9.2 shows graphs of average distances versus network order $N$. Taking advantage of additional enhanced links, both networks $EGIQ(N)$ and $HC(N)$ have reduced values of average distances.

9.2. Network Cost

The cost of a regular network is defined as the product of its diameter and degree. For example, cost of a ring network or the complete graph is proportional to the network size. The costs of $n$-dimensional binary hypercubes and folded hypercubes are $n^2$ and $\frac{in}{2}(n + 1)$, respectively.

We define a new cost factor to evaluate irregular networks with unit incre-
Figure 9.3: The cost of networks $HC_n$, $Q_n$, and $FQ_n$.

mentability, called the \textit{average-distance-degree cost}, which is defined as the product of average distance and average degree. The following property gives the \textit{average degree} of the $n$-dimensional Hamming cube $HC_n$.

\textbf{Property 9.3.} The average degree of $HC_n$, $\text{avgdeg}(HC_n) = \frac{7}{4}n + \frac{1}{4} - 2^{1-n}$.

\textbf{Proof:} With the help of Property 4.7,

\[
\text{avgdeg}(HC_n) = \frac{1}{2^n} \left[ \sum_{j=1}^{n} (n + j)2^{n-j} + 2(2n - 1) \right] \\
= \frac{1}{2^n} \left[ \sum_{i=1}^{n} (2n - i)2^{i} + 2(2n - 1) \right] = \frac{7}{4}n + \frac{1}{4} - 2^{1-n}. \quad \Box
\]

Figure 9.3 shows the network costs of $HC_n$, $Q_n$, and $FQ_n$. It can be seen that the cost of $HC_n$ is less than those of $Q_n$ and $FQ_n$.

By Properties 4.4, 5.2, and the average-distances obtained in the previous sec-
Figure 9.4: The average-distance-degree costs of networks $EGIQ(N)$, $HC(N)$, and $GIQ(N)$.

It can be seen that due to the lower average-distance, both the network costs of $EGIQ(N)$ and $HC(N)$ are less than that of $GIQ(N)$. Although the average-distance of $HC(N)$ is less than that of $EGIQ(N)$, the network cost of $HC(N)$ is larger than that of $EGIQ(N)$. This is because the recursive nature of network $HC(N)$ makes a node $v_i$, where $0 \leq i < 2^n$ and $2^n < N < 2^{n+1}$, have a larger degree. On the contrary, network $EGIQ(N)$ designed from the concept of link replacement eliminates this kind of increase in node degree, which is a trade-off of the recursive property.
9.3. Message Traffic Density

In addition to the diameter and average distance, *message traffic density* \((MTD)\), is another measure of network performance. The \(MTD\) of a network indicates the average number of messages per link per time unit, which reflects the utilization of the network links. A network with lower \(MTD\) has lesser possibility for communication congestion, and hence it could perform more efficient message processing or queuing.

Assume that each node (a sender) in a network sends messages to a distinct node (a receiver) at the same average rate. Thus, on average, the message traveling
distance between the sender and the receiver will be the average distance of the network. Therefore, the factor $MTD$ of a network is $\bar{d}N/E$, where $\bar{d}$ is the average distance, $N$ is the network size, and $E$ is the number of links for message traffic.

The MTD's of networks $HC_n$, $Q_n$, and $FQ_n$ are plotted in Figure 9.5, which have asymptotic values of 0.75, 1, and 0.8, respectively.

The MTD's of networks $EGIQ(N)$, $HC(N)$, and $GIQ(N)$ are plotted in Figure 9.6, which have asymptotic values of 0.8, 0.65, and 1, respectively. Due to the larger number of extra links, the Hamming cube $HC(N)$ has the lowest value MTD among
these networks.

9.4. Summary

The average distance of network defined as the average distance among all pair of nodes, indicates the mean message traversal delay. Taking advantages of enhanced edges and recursive nature, the average distance of the $n$-dimensional Hamming cube $HC_n$ is less than those of binary hypercubes $Q_n$ and folded hypercubes $FQ_n$. Due to additional enhanced links, both the enhanced generalized incomplete hypercube $EGIQ(N)$ and Hamming cube $HC(N)$ of order $N$, where $N$ is not a power of two, have reduced values of average distances, compared to generalized incomplete hypercubes $GIQ(N)$.

We have demonstrated that the average-distance-degree cost of of $HC_n$ is less than those of $Q_n$ and $FQ_n$. Due to the lower average-distance, the costs of $EGIQ(N)$ and $HC(N)$ are less than that of $GIQ(N)$. The message traffic densities of these networks are also computed.
CHAPTER 10

EMBEDDINGS IN HAMMING CUBES

The embedding of a guest network onto a host network provides the geometrical mapping between them. An embedding with lower costs implies that the guest network can be efficiently simulated by the host network. In this chapter, we explore the embeddability of Hamming cubes. Several standard parallel structures are considered as guest networks including Hamiltonian path and cycle, complete binary tree and its variants, and tree machine. The optimal goal of our embedding is set to both the load cost and edge-congestion equal to one. Since there is a trade-off between dilation and expansion, we consider embeddings which provide dilation cost equal to one and expansion is minimum.

10.1. Embedding Hamiltonian Cycles

Binary hypercubes are Hamiltonian because they have Hamiltonian cycles with even length. In a binary hypercube, the sequence of nodes traversed along Gray Codes [61] forms an embedded Hamiltonian cycle. Since there are n! different gray codes each which corresponds to a permutation of the edges-dimensions, an n-dimensional binary hypercube, $Q_n$, can have at least n! different Hamiltonian cycles. However, it has been further shown [66] that a $Q_n$ has $2^n - 3n!$ different Hamiltonian cycles.

\[1\] Two embedded Hamiltonian cycles are said to be different if they have at least one edge in cycle being different.
constructed by the following algorithm.

10.1.1. Hamiltonian Cycles in $Q_n$

Let $D = \{0, 1, \ldots, n - 1\}$ be the set of dimensions of edges in an $n$-dimensional binary hypercube $Q_n$. A pair of nodes is connected by an $i$-dimensional edge if and only if their binary representation of nodes-labels differ at the $(i + 1)$th bit position, counting from the least significant bit as one. A sequence of dimensions, $S$, which determines the traversal of edges in an embedded Hamiltonian cycle is chosen from $D$ and constructed as follows:

**Algorithm Sequencing:** Construct the sequence of dimensions $S$.

begin
1. Arbitrarily choose a dimension $d_1 \in D$ and let $D = D - \{d_1\}$.
2. Let $S_1 = d_1$.
3. For each $i, 2 \leq i \leq n$, choose a dimension $d_i \in D$ and let $D = D - \{d_i\}$.
   Let $S_i = S_{i-1} \star d_i \star S_{i-1}$, where $\star$ indicates the concatenation operation.
4. Let $S = S_n \star d_n$.
end

Given a node $v_1$ and a sequence of dimensions $S_\pi$, an embedded Hamiltonian cycle $C(v_1, S_\pi)$ traverses as one starts from $v_1$ (starting node), and follows the cycle-edges determined by the sequence $S_\pi$. Since there are $2^n$ possible choices of node $v_1$ and $n!$ possible choices of permutation $\pi$, there are at most $2^n n!$ embedded Hamiltonian cycles in the $n$-dimensional binary hypercube, $Q_n$. However, not all such Hamiltonian cycles are different. It has been shown that Algorithm Sequencing can
construct \(2^{n-3}n!\) different embedded Hamiltonian cycles [66].

10.1.2. Hamiltonian Cycles in \(HC_n\)

By definition, a Hamming cube network has two kinds of edges: the \(E_1\)-edges (hypercubic edges) and \(E_2\)-edges (enhanced edges). In the \(n\)-dimensional Hamming cube, \(HC_n\), there exist \(n\)-dimensional \(E_2\)-edge of this form \((v_i, v_j)\) for all nodes \(v_i\). Note that such \(E_2\)-edges have dimension of \(n\). Therefore, the set \(D = \{0, 1, \ldots, n - 1, n\}\) of dimension of edges in \(HC_n\) has one more cardinality than that in \(Q_n\). Obviously, we have \((n + 1)!\) potential choices of permutation \(\pi\). As a consequence, the embedded Hamiltonian cycles in \(HC_n\) constructed by Algorithm Sequencing is given by the following theorem.

**Theorem 10.1.** An \(n\)-dimensional Hamming cube has \(2^{n-3}(n + 1)!\) different Hamiltonian cycles.

![Figure 10.1: An embedding of Hamiltonian cycle \(C(v_0, S_{20n_4})\) in \(HC_4\).](image)

**Example 10.1:** Consider an embedding of a Hamiltonian cycle \(C(v_i, S_\pi)\) in \(HC_4\), where the starting node \(v_i = v_0\). Let the permutation of dimensions \(\pi = 20n_4\). By Algorithm Sequencing, the sequence of dimensions \(S_{20n_4} = 202n_42021202n_42021\). The embedded cycle is \(C(v_0, s_{20n_4}) = (0, 4, 5, 1, 14, 10, 11, 15, 13, 9, 8, 12, 3, 7, 6, 2, 0)\), as shown in Figure 10.1. Note that the ring edges \((v_1, v_{14})\) and \((v_{12}, v_0)\) are of dimen-
137

10.1.3. Hamiltonian Cycles in $HC(N)$, where $2^{k-1} < N < 2^k$ and $k \geq 3$.

It has been shown in [66] that a (generalized) incomplete hypercube of order $N$ has $(k - 2)!$ Hamiltonian paths if $N$ is odd, and $(k - 2)!$ Hamiltonian cycles, if $N$ is even, where $k = \lceil \log N \rceil$. The embedded Hamiltonian cycles $C(v_0, S_\pi)$ in incomplete hypercubes is constructed by applying Algorithm Sequencing with the following modifications:

(1) always select node $v_0$ as the starting node for traversing Hamiltonian cycles or paths.

(2) restrict the permutation $\pi = (p_1, p_2, \ldots, p_n)$ by making $p_1 = n - 1$ and $p_2 = 0$.

(3) remove those nodes in $C(v_0, S_\pi)$ that do not exist in the incomplete hypercube, but exist in the corresponding "complete" binary hypercube.

For the sake of completeness, let us give an example.

**Example 10.2:** Consider an embedding of a Hamiltonian cycle $C(v_0, S_\pi)$ in the incomplete hypercube $GIQ(14)$ of order 14, where $\pi = 3021$. By Algorithm Sequencing, $S_{3021} = 3032303130323031$. $C(v_0, S_{3021}) = (0, 8, 9, 1, 5, 13, 12, 4, 6, 14, 15, 7, 3, 11, 10, 2, 0)$. Since nodes $v_{14}$ and $v_{15}$ do not exist in $GIQ(14)$, we remove them from $C(v_0, S_{3021})$, and the thus required Hamiltonian cycle can be obtained.

For the Hamming cube $HC(N)$ of order $N$, we consider two cases to embed Hamiltonian cycles, depending on the network order: (1) when $2^{k-1} < N < 2^{k-1} + 2^{k-2}$
and \( k \geq 3 \) and (2) when \( 2^{k-1} + 2^{k-2} < N < 2^k \) and \( k \geq 3 \).

10.1.3.1. Case 1: \( HC(N) \) for \( 2^{k-1} < N \leq 2^{k-1} + 2^{k-2} \) and \( k \geq 3 \).

Let \( N = 2^{k-1} + m \), where \( 1 \leq m \leq 2^{k-2} \). The decomposition yields \( HC(N) = \{HC_{k-1}, GIQ(m)\} \), induced by the node-sets \( V(HC_{k-1}) = \{v_\alpha \mid 0 \leq \alpha < 2^{k-1}\} \) and \( V(GIQ) = \{v_\alpha \mid 2^{k-1} \leq \alpha < 2^{k-1} + m\}, \) respectively. Each node \( v_i \) in subgraph \( GIQ(m) \) is linked to two different nodes \( v_{i[1]} \) and \( v_{i[2]} \) in \( HC_{k-1} \) through the \((k-1)\)-dimensional \( E_1 \)-edge and \( n_{k-1} \)-dimensional \( E_2 \)-edge, respectively. Since the Hamming distance \( \rho(v_{i[1]}, v_{i[2]}) = k - 1 \), there exists an \((n_{k-1})\)-dimensional \( E_2 \)-edge between nodes \( v_{i[1]} \) and \( v_{i[2]} \). Thus, the following property is immediately obtained.

**Property 10.1.** Let nodes \( v_\alpha \) and \( v_\beta \) be two nodes in the \((k-1)\)-dimensional Hamming cube \( HC_{k-1} \) for \( k \geq 3 \). Node \( v_\alpha \) is linked to \( v_\beta \) through an \( n_{k-1} \)-dimensional \( E_2 \)-edge, i.e. \( \alpha = \beta^{(k-1)} \). Let \( v_\gamma \) be a node in Hamming cube \( HC_k \) such that \( \alpha = \gamma^{(k-1)} \) and \( \beta = \gamma^{(k)} \). In \( HC_k \), there exists a path \( P = (v_\alpha, v_\gamma, v_\beta) \) of length 2 between nodes \( v_\alpha \) and \( v_\beta \), which bypasses node \( v_\gamma \).

By Theorem 10.1 and Property 10.1, we can embed Hamiltonian cycles in Hamming cubes of Case 1. We first apply Algorithm Sequencing, to construct the Hamiltonian cycle \( C(v_i, S_\pi) \) in the subgraph \( HC_{k-1} \) by letting element \( p_1 = n_{k-1} \) in the permutation \( \pi = (p_1, p_2, \ldots, p_{k-1}) \), and choosing the other elements \( p_i \)'s are arbitrarily from the set \( D = \{0, 1, \ldots, k - 2\} \). Note that the starting node \( v_i \) can be any node in \( HC_{k-1} \). The resulting sequence of dimensions corresponding to \( \pi \) has the form \( S_{n_{k-1}p_2\ldots p_{k-1}} = n_{k-1}p_2n_{k-1}p_3n_{k-1}p_4n_{k-1}\ldots \). Let \( C(v_i, S_{n_{k-1}p_2\ldots p_{k-1}}) = i a_2 a_3 \ldots, a_{2^{k-1}}i \). We search those pairs of nodes in the cycle such that they are linked through the \( n_{k-1} \)-dimensional \( E_2 \)-edges, and then according to Property 10.1, we appropriately insert the nodes of the subgraph \( GIQ(m) \) between them, where
Since there are $2^{k-1}$ possible choices of the starting node $v_i$ and $(k-1)!$ possible choices of the permutation $\pi$, the subgraph $HC_{k-1}$ has $2^{k-4}(k-1)!$ different Hamiltonian cycles, for $k \geq 4$. For each instance of embedded Hamiltonian cycles in $HC_{k-1}$, we expand the cycle edges, which are $n_{k-1}$-dimensional $E2$-edges, as paths with length two to include the nodes of the subgraph $GIQ(m)$. Furthermore, by Property 10.1, the edges used in the expanded paths are different from those used in the Hamiltonian cycle in $HC_{k-1}$. Therefore, we have the following lemma.

**Lemma 10.1.** The Hamming cube of order $N$, where $2^{k-1} < N < 2^{k-1} + 2^{k-2}$ and $k \geq 4$, has $2^{k-4}(k-1)!$ different Hamiltonian cycles.

**Example 10.3:** Consider $HC(11) = \{HC_3, GIQ(3)\}$, where $V(HC_3) = \{v_{\alpha} \mid 0 \leq \alpha \leq 7\}$ and $V(GIQ) = \{v_8, v_9, v_{10}\}$. Let permutation $\pi = n_320$ and a starting node $v_i = v_6$. Then $S_{n_320} = n_32n_30n_32n_30$. The embedded Hamiltonian cycle in $HC_3$ is $C(v_6, S_{n_320}) = (6, 1, 5, 2, 3, 4, 0, 7, 6)$, as shown in Figure 10.2(a). We insert each of nodes $v_8$, $v_9$, and $v_{10}$ into the pairs of nodes $(0, 7)$, $(6, 1)$, and $(5, 2)$, respectively. The Hamiltonian cycle corresponding to $C(v_6, S_{n_320})$ is $(6, 9, 1, 5, 10, 2, 3, 4, 0, 8, 7, 6)$,
as shown in Figure 10.2(b).

10.1.3.2. Case 2: $HC(N)$ for $2^{k-1} + 2^{k-2} < N < 2^k$ and $k \geq 3$.

Consider $N = 2^{k-1} + m$, where $2^{k-2} < m < 2^{k-1}$, and decomposition $HC(N) = \{HC_{k-1}, GIQ(m)\}$. For the embedding of Hamiltonian cycles, we divide this case into two subcases depending on whether $m$ is an even number or not.

Case 2.1: When $m$ is even.

In this subcase, the subgraph $GIQ(m)$ induced by the node-set $V(GIQ(m)) = \{v_\alpha \mid 2^{k-1} \leq \alpha < 2^{k-1} + m\}$ is an incomplete hypercube having an even number of nodes. Therefore, we can use the method introduced at the beginning of Section 10.1.2 to embed Hamiltonian cycles in $GIQ(m)$, which is formally described below:

Let $Q'_k$ be the corresponding “complete” binary hypercube of $GIQ(m)$, given by $V(Q'_k) = \{v_\alpha \mid 2^{k-1} \leq \alpha < 2^k\}$. By Algorithm Sequencing, we construct the embedded Hamiltonian cycle $C(v_{2^{k-1}}, S_\pi)$ in $Q'_k$, where $p_1 = k - 2$ and $p_2 = 0$ in the permutation $\pi = (p_1, p_2, \ldots, p_{k-1})$. Then, the sequence of dimensions will be $S_\pi = (k-2)0(k-2)p_3(k-2)0(k-2)\ldots p_{k-1}$. The Hamiltonian cycle in $GIQ(m)$, $C' = (2^{k-1}a_2a_3\ldots a_m2^{k-1})$, is obtained by removing the nodes $v_i \in V(Q'_k) - V(GIQ(m))$ from $C(v_{2^{k-1}}, S_\pi)$. Let $p' = p_{k-1}$ be the dimension of the $E1$-edge through which the last node $a_m$ in $C'$ traverses back to the starting node $2^{k-1}$.

Now, let us consider another induced subgraph $HC_{k-1}$ in the decomposition of $HC(N)$. We embed Hamiltonian cycles in it according to the embedded $C'$ in $GIQ(m)$, such that they can be joined together to form the embedded Hamiltonian cycle in $HC(N)$.

Since the subgraph $HC_{k-1}$ is the $(k - 1)$-dimensional Hamming cube, we can construct the embedded Hamiltonian cycle, denoted as $C(v_i, S_\pi)$, by the method
presented in Section 10.1.1. We restrict the starting node \( v_i = v_0 \), and permutation \( \pi = p_1 p_2 \ldots p' \). Note that \( p' \) is the same dimension of edge as used to link the last node to the starting node \( 2^k - 1 \) in the embedded \( C' \) of \( GIQ(m) \). Thus, the corresponding \( S_{p_1 p_2 \ldots p'} = p_1 p_2 p_1 \ldots p' \) and \( C(v_0, S_{p_1 p_2 \ldots p'}) = (0, b_2, b_3, \ldots, b_{2^{k-1}}, 0) \). In the following, let us join \( C' \) with \( C(v_0, S_{p_1 p_2 \ldots p'}) \) to form the embedded Hamiltonian cycle in \( HC(N) \).

Without loss of generality, we can reverse the order of nodes in \( C' = (2^{k-1} a_2 a_3 \ldots a_m 2^{k-1}) \) as \( C'' = (2^{k-1} a_m a_{m-1} \ldots a_2 2^{k-1}) \). Then we link nodes 0 and \( b_{2^{k-1}} \) in \( C(v_0, S_{p_1 p_2 \ldots p'}) \) to nodes \( 2^{k-1} \) and \( a_m \), respectively, by the \( (k - 1) \)-dimensional \( E1 \)-edges. The new combined Hamiltonian cycle \( C_{HC(N)} = (0, b_2, \ldots, b_{2^{k-1}}, a_m, a_{m-1}, \ldots, 2^{k-1}, 0) \) is the embedded Hamiltonian cycle in \( HC(N) \). Since there are \( (k - 1)! \) possible choices of the permutation \( \pi \) in \( C(v_0, S_{p_1 p_2 \ldots p'}) \) and \( \left( \lceil \log(N - 2^{k-1}) \rceil - 2 \right)! \) possible choices of the permutation \( \pi \) in \( C' \), we have the following lemma.

**Lemma 10.2.** The Hamming cube of even order \( N \), where \( 2^{k-1} + 2^{k-2} < N < 2^k \) for \( k \geq 3 \), has \( (k - 1)! \left( \lceil \log(N - 2^{k-1}) \rceil - 2 \right)! \) different Hamiltonian cycles.

![Figure 10.3: The embedding of Hamiltonian cycles in \( HC_3, GIQ(6) \), and \( HC(14) \).](image)

**Example 10.4:** Consider \( HC(14) = \{ HC_3, GIQ(6) \} \). The embedded Hamiltonian cycles in the subgraphs \( HC_3 \) and \( GIQ(6) \) are \( C(v_0, S_{0 m_{11}}) = (0, 1, 6, 7, 5, 4, 3, 2, 0) \) and \( C' = (8, 12, 13, 9, 11, 10, 8) \), shown in Figures 10.3(a) and (b), respectively. Joined by the edges \((v_8, v_0)\) and \((v_2, v_{10})\), the corresponding Hamiltonian cycle is \( C_{HC(N)} = \).
as shown in Figure 10.3(c).

Case 2.2: When $m$ is odd.

With the help of the embedding in the previous subcase and Property 10.1, we can embed a Hamiltonian cycle in the Hamming cube $HC(N)$, where $N = 2^{k-1} + m$, $2^{k-2} < m < 2^{k-1}$, and $m$ is an odd number. Let $m = m' + 1$. Then $HC(N - 1) = \{HC_{k-1}, GIQ(m')\}$. We first embed the Hamiltonian cycle $C_{HC(N-1)}$ in $HC(N - 1)$ by the method in Case 2.1, for which the embedded of $C(v_0, S_{n_{k-1}p_1p_3...p'})$ in $HC_{k-1}$ needs to be constructed. Note that the element $p_1 = n_{k-1}$ in permutation $\pi$ is different from an arbitrary permutation for the case when $m$ is even. The embedding of $C_{HC(N-1)}$ then includes node $v_{N-1}$ by expanding the edge $(v_{(N-1)^{i-1}}, v_{(N-1)^{i+2}})$ as a path $(v_{(N-1)^{i-1}}, v_{N-1}, v_{(N-1)^{i+2}})$ according to Property 10.1.

Due to the restriction on the first element in the permutation $\pi$ and by Lemma 10.2, we have the following lemma.

**Lemma 10.3.** The Hamming cube of odd order $N$, where $2^{k-1} + 2^{k-2} < N < 2^k$ for $k \geq 3$, has $(k - 2)!([\log(N - 2^{k-1})] - 2)!$ different Hamiltonian cycles.

![Figure 10.4: An embedding of Hamiltonian cycles in $HC(15)$.](image)

**Example 10.5:** Figure 10.4(a) shows the embedding of $C_{HC(14)} = (0, 7, 6, 1, 3, 4, 5, 2, 10, 11, 9, 13, 12, 8, 0)$ with $C(v_0, S_{n301}) = (0, 7, 6, 1, 3, 4, 5, 2, 0)$. Node $v_{14}$ is included by the
path \((v_1, v_14, v_6)\), as shown in Figure 10.4(b).

By Lemmas 10.1, 10.2, and 10.3, we obtain

**Theorem 10.2.** The Hamming cube of order \(N\), where \(2^{k-1} < N < 2^k\) for \(k \geq 3\), has \((k - 2)!\left(\lceil \log(N - 2^{k-1}) \rceil - 2\right)!\) different Hamiltonian cycles.

In the Hamming cube \(HC(N)\), we can easily embed Hamiltonian cycles of any order \(2 \leq N \leq 4\). Therefore, Hamming cubes are *pancyclic*.

### 10.2. Embedding Complete Binary Trees and Their Related

Both the complete binary tree and binary hypercube are bipartite graphs. A node of the binary hypercube has even parity if its binary representation has an even number of one bits; otherwise, it has an odd parity. An \(n\)-dimensional binary hypercube has \(2^n-1\) even parity nodes and \(2^n-1\) odd parity nodes. In the binary tree, the nodes at the even (or odd) levels form a parity node-set. Therefore, it can be shown that the complete binary tree \(CBT(n - 1)\) of height \(n - 1\) consisting of \((2^n - 1)\) nodes is not a subgraph of the \(n\)-dimensional binary hypercube, \(Q_n\). However, the \(2^n\)-node *two-rooted* complete binary tree is a subgraph of \(Q_n\) [12, 34, 54]. Thus \(CBT(n - 1)\) can be embedded into \(Q_n\) with dilation two, and \(CBT(n - 1)\) is a subgraph of \(Q_{n+1}\).

As shown in Section 4.3, an \(n\)-dimensional Hamming cube can be decomposed into two induced disjoint subgraphs, an \((n - 1)\)-dimensional Hamming cube and an \((n - 1)\)-dimensional binary hypercube, denoted as \(HC_n = \{HC_{n-1}, Q_{n-1}\}\). Each subgraph is induced by the vertex-subset \(V' = \{v_\alpha \mid 0 \leq \alpha < 2^{n-1}\}\) and \(V'' = \{v_\alpha \mid 2^{n-1} \leq \alpha < 2^n\}\), respectively. By the recursive property of networks, the subgraphs \(HC_{n-1}\) and \(Q_{n-1}\) can be further decomposed as \(HC_{n-1} = \{HC_{n-2}, Q_{n-2}\}\) and \(Q_{n-1} = \{Q^1_{n-2}, Q^2_{n-2}\}\). Therefore, the decomposition \(HC_n = \{HC_{n-2}, Q_{n-2}, Q^1_{n-2}, Q^2_{n-2}\}\).
Note that the node in each subgraph of \( HC_n \) has the labels \((00*_{n-2})\), \((01*_{n-2})\), \((10*_{n-2})\), and \((11*_{n-2})\), where * means don’t care.

Consider the case when a node \( v_i \in Q_{n-2} \), where label \( i = (01*_{n-2}) \). By the definition of Hamming cubes, \( v_i \) is linked to nodes \( v_{i[1]} \) and \( v_{i[n]} \) (or \( v_{i(n)} \)) through the \((n - 1)\)-dimensional \( E1 \)-edge and \( n \)-dimensional \( E2 \)-edge, respectively. Since labels \( v_{i[1]} = (11*_{n-2}) \) and \( v_{i} = (10*_{n-2}) \), nodes \( v_{i[1]} \in Q^2_{n-2} \) and \( v_{i} \in Q^1_{n-2} \). A node \( v_i \in HC_{n-2} \) also has similar property, as stated below:

**Property 10.2.** Consider the \( n \)-dimensional Hamming cube, \( HC_n = \{HC_{n-2}, Q_{n-2}, Q^1_{n-2}, Q^2_{n-2}\} \) for \( n \geq 2 \), in which the subgraphs are induced by the vertex-subsets

\[
V^1 = \{v_\alpha \mid 0 \leq \alpha < 2^{n-2}\}, \quad V^2 = \{v_\alpha \mid 2^{n-2} \leq \alpha < 2^{n-1}\}, \quad V^3 = \{v_\alpha \mid 2^{n-1} \leq \alpha < 2^{n-1} + 2^{n-2}\}, \quad \text{and} \quad V^4 = \{v_\alpha \mid 2^{n-1} + 2^{n-2} \leq \alpha < 2^n\}.
\]

A node \( v_i \in Q_{n-2} \) (or \( HC_{n-2} \)) is linked to two nodes \( v_{i[1]} \in Q^1_{n-2} \) and \( v_{i[n]} \in Q^2_{n-2} \), through the \((n - 1)\)-dimensional \( E1 \)-edges and the \( n \)-dimensional \( E2 \)-edges, respectively.

![Figure 10.5: The embedding of complete binary tree of height 4 in \( HC_5 \).](image-url)

By Property 10.2, we can embed the complete binary tree \( CBT(n - 1) \) of height
Let node $v_0$ be the root in the single node tree. Let $CBT(n-2)$ be the embedded complete binary tree in $HC_{n-1}$, having the leaves of nodes in the set $V^2$ defined in Property 10.2. The embedding $CBT(n-1)$ in $HC_n$ grows from $CBT(n-2)$ by making the nodes in the set $V'' = V^3 \cup V^4$ the children of the leaves of $CBT(n-2)$ through the $(n-1)$-dimensional $E_1$-edge and $n$-dimensional $E_2$-edge.

By Property 4.7, the two-dimensional Hamming cube $HC_2$ consisting of nodes $v_0, v_1, v_2, v_3$ is a complete graph $K_4$. The root of the embedded $CBT$ can be any one of them.

Figure 10.5 shows the embedding of $CBT(4)$ rooted at $v_0$ in $HC_5$. The label on an edge indicates its dimension. By our tree construction and Property 10.2, we have the following theorem.

**Theorem 10.3.** The $(2^n - 1)$-node complete binary tree is a subgraph of the $n$-dimensional Hamming cube having $2^n$ nodes.

Note that the subgraph of $HC_n$ induced by the set $V'' = V^3 \cup V^4$ forms an $(n-1)$-dimensional binary hypercube. In addition, an embedded complete binary tree with a large value of height is recursively constructed from the ones with smaller values of height. Therefore, the nodes at each level $j$ for $0 \leq j < n$ of the embedded complete binary tree $CBT(n-1)$ in $HC_n$ are connected as a $j$-dimensional binary hypercube $Q_j$ which has the nodes $V(Q_j) = \{v_i \mid 2^j \leq i < 2^{j+1}\}$. We call such a network architecture a tree-cube. Figure 10.6 shows the embedding of tree-cube $TC(3)$ of height 3 in $HC_4$. In the figure, the broken lines at each level of the tree indicate the additional hypercube edges existing in $HC_4$. With these edges, the nodes at each level $i$ for $0 \leq i \leq 3$ form the $i$-dimensional binary hypercube. Due to the existence of a tree cube, several variants of complete binary trees with additional links
between the nodes at the same level can be embedded in the $n$-dimensional Hamming cube.

For example, The hypertree structure [43] adds some additional edges, called hyper-edges, to a complete binary tree. The additional links at each level are chosen to be a subset of the hypercube (Section 2.4). Since the embedded tree-cube structure has all the hypercube links at each level of nodes, the hypertree $HT(n-1)$ of height $n-1$, is a subgraph of tree-cube $TC(n-1)$ of height $n-1$, and hence a subgraph of the $n$-dimensional Hamming cube, $HC_n$. Figure 10.7 shows the embedding of $HT(4)$ in $HC_5$. In the figure, the dashed lines indicate the additional hyper-edges.

A full-ringed (binary) tree [33] is a complete binary tree in which all nodes at each level are connected as a ring. If the ring-edges between pairs of nodes having the same parent are omitted, then the tree is a half-ringed tree. With the help of the embedded tree-cube in Hamming cubes, and the fact that the binary hypercubes
Figure 10.7: The embedding of hypertree $HT(4)$ of height 4 in $HC_5$.

are Hamiltonian, the full-ringed tree (also half-ringed tree) of height $n - 1$ can be embedded in the $n$-dimensional Hamming cube. Figures 10.8 and 10.9 show the embeddings of full-ringed tree and half-ringed tree of height 3 in $HC_4$. The dashed lines in the figure are those additional ring edges.

10.3. Embedding Tree Machine

A tree machine, $TM(n)$, of dimension $n$ consists of two complete binary trees of height $n$, upper tree and lower tree, which are connected back to back along the common leaves. A $TM(n)$ contains $(3 \cdot 2^n - 2)$ nodes and $(2^{n+2} - 2)$ edges. It can be embedded in the $(n + 2)$-dimensional binary hypercube $Q_{n+2}$ with expansion approximately equal to $\frac{4}{3}$, and dilation one [34]. It has been also shown [57] that $TM(n)$ can be embedded in the (generalized) incomplete hypercube of order $3 \cdot 2^n$ or
Figure 10.8: The embedding of full-ring tree of height 3 in $HC_4$.

$(3 \cdot 2^n + 2^{n-1})$ with both dilation and edge congestion equal to two.

In this section, we will show that a $TM(n)$ is a subgraph of the Hamming cube of order $(3 \cdot 2^n + 2^{n-1})$. Such an embedding has dilation one and expansion approximately equal to $\frac{7}{8}$.

For the embedding of a tree machine in the Hamming cube, we view the structure of the tree machine from a different angle. In $TM(n)$, the $2^n$ common leaves and their $2^n$ parents (half of them in the upper tree and another half in the lower tree) form $2^{n-1}$ number of two-dimensional binary hypercubes $Q_2$, called building blocks. These building blocks are then connected by the upper and lower trees of height $n - 1$, one less than the original. Note that the leaves of these two new trees are now the parents of cornerwise nodes in the building blocks. When the dimension of the tree machine increases, say from $n$ to $n + 1$, the number of building blocks becomes double, from $2^{n-1}$ to $2^n$. Thus, we need $2^n$ new leaves for each upper and lower tree to connect the whole set of $2^n$ building blocks.

In the Hamming cube of $HC(3 \cdot 2^n + 2^{n-1})$ for $n \geq 3$, each node label has
Figure 10.9: The embedding of half-ring tree of height 3 in $HC_4$.

length $n + 2$. According to the first and second lowest bits of node labels, we can decompose $HC(3 \cdot 2^n + 2^{n-1})$ into $(3 \cdot 2^{n-2} + 2^{n-3})$ number of building blocks, $Q^i_2$, for $0 \leq i \leq 3 \cdot 2^{n-2} + 2^{n-3} - 1$. Since the upper and lower trees in $TM(n)$ are symmetric along their common leaves, we can concentrate on any one tree without loss of generality.

Let nodes $v_3$ and $v_0$ in $Q^0_2$ be the roots of the upper and lower trees in $TM(n)$, respectively. The root $v_3$ has the children $v_1$ and $v_4$, and $v_0$ has the children $v_2$ and $v_7$, both through the 1-dimensional $E1$-edge and the $n_3$-dimensional $E2$-edge.

In the embedded $TM(n)$ for $n \geq 3$, a nonterminal (or internal) node $v_i$ at level $j$ for $1 \leq j \leq n - 3$, of the upper tree has the left child $v_{i(2^j+2)}$ and right child $v_{i(2^j+1)}$ linked through a $(2j+1)$-dimensional $E1$-edge and $n_{2j+2}$-dimensional $E2$-edge, respectively. So far, we can construct $n - 1$ levels of the upper tree. The remaining step that needs to be done is to construct the leaves of the upper tree which are those parents of cornerwise nodes in the building blocks.

We divide the nodes at level $n - 2$ into two sets, $V'$ and $V''$. The set $V'$ contains the first $2^{n-3}$ nodes from the left, while the set $V''$ contains the remaining $2^{n-3}$ nodes.
on that level. A node $v_i \in V'$ has two leaves $v_{i[n+1]}$ and $v_{i[n+1]}$ linked through the $n$-dimensional $E1$-edge and $(n_{n+1})$-dimensional $E2$-edge, respectively. While a node $v_i \in V''$ has two leaves $v_{i[n+2]}$ and $v_{i[n+2]}$ linked through the $n+1$-dimensional $E1$-edge and $n_{n+2}$-dimensional $E2$-edge, respectively. Thus, we can construct the entire upper tree.

By the preceding method, the lower tree rooted at node $v_0$ can also be constructed. The common leaves for both trees are then determined by the parent nodes of building blocks, which are the leaves of upper and lower trees, through the $E1$-edges of dimensions 0 and 1.

![Diagram](image)

Figure 10.10: The embedding of $TM(3)$ in $HC(28)$.

Figure 10.10 shows the embedding of $TM(3)$ in $HC(28)$. There are four building blocks formed by the 0- and 1-dimensional $E1$-edges. Figure 10.11 shows the upper tree in $HC(28)$. The figure includes all nodes of $HC(28)$, but omits the edges which are not used in the tree. From Figures 10.10 and 10.11, we can clearly see the
Figure 10.11: The upper tree of $TM(3)$ in $HC(28)$.

geometric relation of four building blocks ($Q_2$'s) in $HC(28)$. Figure 10.12 shows the embedded $TM(4)$ in $HC(56)$.

10.4. Summary

In this chapter, we have optimally embedded Hamiltonian path and cycle, complete binary tree and its variants, and tree machine into Hamming cubes with the costs of load, edge-congestion, and dilation all equal to one, and with the minimum expansion. The embedding of Hamiltonian cycles conclude that Hamming cubes are pancyclic networks, that is the Hamiltonian cycles of all lengths can be embedded.

The complete binary tree is a subgraph of the Hamming cube with the same size. A tree-cube is a complete binary tree in which the nodes at each level are connected as a binary hypercube. The tree-cube is also a subgraph of Hamming cube with the same size. Therefore, several variants of the complete binary tree such as X-tree, Hypertree, full-ringed tree, and half-ringed tree can be embedded into Hamming
Figure 10.12: The embedding of $TM(4)$ in $HC(56)$.

cubes with the expansion equal to one.

Tree machines can be embedded in the generalized incomplete hypercubes with expansion approximately equal to one, and both dilation and edge congestion equal to two. With the same costs, the tree machine can also be embedded into Hamming cubes. However, it has been explored that the tree machines can be embedded into Hamming cubes with dilation of one and expansion of $\frac{7}{6}$. Keeping the same expansion of $\frac{7}{6}$, the embedding of tree machines in the generalized incomplete hypercubes have dilation and edge congestion equal to two. Table 10.1 summarizes the embedding results of the hypercube-family of networks and Hamming cubes.
CHAPTER 11

CONCLUSIONS

In this dissertation, we have introduced the Hamming group graphs as a network model. A Hamming-group graph $HGG = (V, \Omega)$ is a transformation graph for which set $V = \{ i | 0 \leq i < 2^n \text{ and } n \geq 1 \}$ is the Hamming group containing all combinations of binary bits of length $n$. A generator $\omega \in \Omega$, which is a binary string of length $n$, acts on the Hamming group by the bitwise Exclusive-OR operation. The Hamming group graphs are the supergraphs of binary hypercubes and have the properties of vertex-symmetry and conditional edge-symmetry.

We have also defined incremental Hamming graphs, which provide unit expansion capability. It is found that the incremental Hamming graph model can generate fertile supergraphs of (generalized) incomplete hypercubes. From our network models, we derive two new families of networks with the variant generators, called the Hamming cubes and enhanced generalized incomplete hypercubes.

We have shown that the Hamming cube is an attractive hypercube-based network with many desirable properties. These networks can recursively grow from the existing ones with the increment of one node at a time, have half of logarithmic diameter and are easily decomposable. For Hamming cubes, simple deterministic self-routing scheme has been designed, which always finds a shortest path of length at most the diameter of the network. The Hamming cube is shown to be an optimally fault-tolerant network from the view point that its node-connectivity is equal to the minimum degree.

Compared to the binary hypercube $Q_n$, the $n$-dimensional Hamming cube $HC_n$
has $2^n - 2$ extra edges, but it provides better fault-tolerance (node-connectivity is $n + 1$) and smaller diameter, equal to $\left\lceil \frac{n}{2} \right\rceil$. Note here that the folded hypercube $FQ_n$ has also identical diameter and node-connectivity as of $HC_n$, at the cost of only $2^{n-1}$ extra edges than in $Q_n$. However, the fact that Hamming cubes are recursively constructible with unit incrementability, makes them more attractive than $FQ_n$. The recursive nature is also an advantage over supercubes. Hamming cubes of arbitrary order are at least biconnected, which implies that they are more fault-tolerant than incomplete hypercubes of the same size.

We have further shown that Hamming cubes exhibit very good performance in terms of fault-tolerance and reliability measures. For example, the $n$-dimensional Hamming cube ($HC_n$) is strongly resilient, whose $n$-fault diameter is its diameter plus one, i.e. $\left\lceil \frac{n}{2} \right\rceil + 1$. Moreover, $HC_n$ has been analyzed by the indicators such as two-terminal (or path) reliability and vulnerability. Since Hamming cubes are recursively defined, there exist additional disjoint paths between two smaller labeled nodes. This implies that the reliability and fault-tolerance of $HC_n$ are at least as good as those of $FQ_n$, and better than those of $Q_n$. Taking advantages of enhanced edges and recursive nature, the average distance, average-distance-degree cost, and message traffic density of $HC_n$ are all less than those of $Q_n$ and $FQ_n$. Furthermore, $HC_n$ has constant vulnerability and can be laid out in an $O(N \times N)$ square, where $N = 2^n$. We have designed a testing algorithm for the injured $HC_n$ having some faulty components (nodes or links). Our diagnostic algorithm can identify up to $n+1$ faulty processors.

In this dissertation, we have proposed another new family of hypercube-based networks, called enhanced generalized incomplete hypercubes. This family also belongs to the incremental Hamming graphs with the variant generator set. By showing
that enhanced incomplete hypercubes and folded hypercubes are the enhanced general-
ized incomplete hypercubes of special orders, these networks can be viewed as
"enhanced" generalized incomplete hypercubes [72, 26] with extra links or "general-
ized" folded hypercubes [36] with incrementability of one.

An enhanced generalized incomplete hypercube (EGIQ) can be viewed as an
"enhanced" incomplete hypercube with extra links or a "generalized" folded hyper-
cube with incrementability of one. This proposed family of networks also has half
of logarithmic diameter. Simple deterministic routing schemes are designed, having
path lengths bounded by the diameter. Due to additional enhanced links, both the
networks EGIQ(N) and HC(N) for an arbitrary order N, have improved values of
network parameters compared to the existing incomplete hypercubes.

Two types of broadcasting primitives — spanning trees ST’s and multiple span-
nning trees MUST’s — have been constructed for generalized incomplete hypercubes,
Hamming cubes, and enhanced generalized incomplete hypercubes. The time com-
plexity by using these broadcasting trees are analyzed by one-port and all-port com-
munication models. For all communications, the source node for data broadcasting
can be an arbitrary node in the network. The ST’s in the above three networks
have been constructed and they have height bounded by the diameters of networks.
For example, the height of \( ST_{HC_n} \) in \( HC_n \) is half of that in \( Q_n \). Due to the half
of logarithmic diameter, the spanning tree \( ST_{HC(N)} \) in a Hamming cube \( HC(N) \) of
an arbitrary \( N \) and the spanning tree \( ST_{EGIQ(N)} \) in enhanced generalized incomplete
hypercube \( EGIQ(N) \) can speed up the time complexity for all-port communication,
faster than that in a generalized incomplete hypercube \( GIQ(N) \). Under the one-port
communication, the broadcasting time by using \( ST_{EGIQ(N)} \) is less than that by using
\( ST_{HC(N)} \).
The MUST in $HC_n$, $MUST_{HC_n}$, consists of $(n + 1)$ EDST’s and has the height of $\left[\frac{n}{2}\right] + 1$. Compared to $Q_n$, $MUST_{HC_n}$ has one more EDST’s which can speed up the data broadcasting in all-port communication, and its height is half which reduces the communication delay in one-port communication and the size of pipeline in all-port communication. The $MUST_{GIQ(n)}$ in $GIQ(N)$ has been constructed to have optimal number, equal to the minimum degree, edge-disjoint spanning trees. The technique for constructing the $MUST_{GIQ(n)}$’s can be modified to construct the MUST’s in enhanced generalized incomplete hypercubes and Hamming cubes. By the additional $E2$-edges in $HC(N)$, one more EDST can be constructed in the $MUST_{HC(N)}$. However, because of link-replacement, the broadcasting time by using $MUST_{EGIQ(N)}$ is almost the same as that of $MUST_{GIQ(N)}$.

We have explored the embeddability of Hamming cubes. Several standard parallel structures including Hamiltonian path and cycle, complete binary tree and its variants, and tree machine can be optimally embedded into Hamming cubes with the costs of load, edge-congestion, and dilation equal to one, and expansion minimum.

Due to bipartiteness of incomplete hypercubes, a Hamiltonian cycle of odd length cannot be embedded with dilation of one. Using the additional enhanced edges in Hamming cubes, the Hamiltonian cycles of all lengths can be embedded as subgraphs.

It is known that the complete binary tree is not a subgraph of binary hypercubes or (generalized) incomplete binary trees of the same size. The complete binary tree is a subgraph of Hamming cube with the same size. Such an embedding has expansion of one. This a clear advange of the Hamming cubes. Furthermore, X-trees, Hypertrees, full-ringed and half-ringed binary trees are subgraphs of Hamming cubes.

Tree machines can be embedded in the generalized incomplete hypercubes with expansion approximately equal to one, and both dilation and edge congestion equal
to two. However, the tree machines can be embedded into Hamming cubes with dilation of one and expansion of $\frac{7}{6}$. With the same expansion of $\frac{7}{6}$, the embedding of tree machines in the generalized incomplete hypercubes have dilation and edge congestion equal to two. This is another advantageous embedding property of Hamming cubes.

10.1. Future Work

Some remarks are made here on the future development of our network model and the Hamming cube network. As shown in Section 2.2, the incremental Hamming graphs are derived from the consideration of unit incrementability, however, in trade-off of the network regularity. The problem that designs a regular hypercube-like network with the expansion of smaller order is still open. The investigations of communication and embedding for the faulty networks is another direction.
<table>
<thead>
<tr>
<th>(HC(n))</th>
<th>(T(n))</th>
<th>(T(n))</th>
<th>(T(n))</th>
<th>(T(n))</th>
<th>(T(n))</th>
<th>(T(n))</th>
</tr>
</thead>
<tbody>
<tr>
<td>(D = EC = \tau) &amp; (D = EC = \tau) &amp; (D = EC = \tau) &amp; (D = EC = \tau) &amp; (D = EC = \tau) &amp; (D = EC = \tau)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(H(n)) &amp; (H(n)) &amp; (H(n)) &amp; (H(n)) &amp; (H(n)) &amp; (H(n))</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\text{Hypercube}) &amp; (\text{Hypercube}) &amp; (\text{Hypercube}) &amp; (\text{Hypercube}) &amp; (\text{Hypercube}) &amp; (\text{Hypercube})</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(G(n)) &amp; (G(n)) &amp; (G(n)) &amp; (G(n)) &amp; (G(n)) &amp; (G(n))</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\text{Hamming Cube}) &amp; (\text{Hamming Cube}) &amp; (\text{Hamming Cube}) &amp; (\text{Hamming Cube}) &amp; (\text{Hamming Cube}) &amp; (\text{Hamming Cube})</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 10.1: Embedding comparisons of hypercube-based networks.
BIBLIOGRAPHY


