Interconnection Topologies and Routing for Parallel Processing Systems

Gabriele Kotsis
Institut f"ur Angewandte Informatik
und Informationssysteme
Universit"at Wien
Lenaugasse 2/8
A - 1080 Wien
Austria

Office Phone Number: +43 1 408636614

Fax: +43 1 408 04 50

Internet: gabi@ani.univie.ac.at

This work was published in the ACPC (Austrian Center for Parallel Computation) Technical Report Series, ACPC/TR 92-19, 1992.

Contents

Preface

"ut desint vires
tamen est laudanda voluntas"
Ovid

A couple of months ago, I was asked the following puzzle by A. Ferscha, a staff member of the *Institut für Statistik und Informatik* at the University of Vienna to find a graph on 16 nodes, with the restriction that every node has at most 4 edges so that the number of edges that must be passed when going from one node to another becomes minimum. Since I have always been interested in puzzles, I started at once drawing a lot of graphs, and soon I have found one, where the maximum distance between all pairs of nodes was 3. But was this the minimum? How could I be sure, that there are no better graphs? These questions set up my interest in the field of graph theory, especially in the problem of finding graphs with minimum diameter (which is defined as the maximum distance between any two nodes). I have found some articles, dealing with similar problems (maximizing the number of nodes in a graph of given diameter), and in [Erdo 80] I finally found the proof, that there is no graph of 16 nodes, with 4 edges per node, and a diameter less than 3.

But what has this problem of graph theory to do with computer science? This question can be answered when analysing the origin of the puzzle. The *Institut für Statistik und Informatik* is equipped with a transputer based parallel processing system consisting of 16 processors, each having 4 communication links to interconnect processing elements. In the above puzzle, the 16 nodes represent the processing elements, and the edges correspond to the links in the transputer. The problem was to find an *interconnection topology* for the transputer, which enables shortest possible communication paths among the processing elements (diameter in the network).

Encouraged by the success in searching for an optimum topology, I spent more time in searching topologies and read a lot of articles on related topics. When studying the literature, I discovered, that there is an enormous amount of topologies proposed for *interconnection networks*, but there is a lack of a detailed survey and comparison of topologies. So I decided to write my thesis about this topic.

I found much support by A. Ferscha, who helped me in finding literature and always found time to discuss the problems, that occured when writing this work. I thank him very much. I would also like to thank Prof. G. Haring, who was so kind to assess my work. I spent a lot of time at the institute doing figures and calculations for the tables, and would like to thank all the staff for supporting me in my work. Special thanks are expressed to M. Johnson, who did a lot of thinking about my topology search programs written in Pascal (finally rewriting them to C), and to M. Schmiedl, who always helped me patiently with printer problems.

Chapter 1

Introduction

The growing demand for more computing power at increasing speed in many scientific and engineering applications made it necessary to develop advanced computer architectures based on the concept of parallel processing. In general a parallel computer system consists of various processing and memory units and other (shared) resources. A critical issue in design and analysis of parallel systems is the way in which the system components are connected together, since this *interconnection network* determines the performance of the whole system [Bhuy 87].

A wide spectrum of parallel systems exists nowadays[1], some of them have been designed for specific applications, others are so called general purpose architectures. In order to specify the demands posed on an interconnection network, we first must classify the parallel architectures, because different architectural concepts reflect in different demands. The most famous classification scheme was proposed by Flynn [Flyn 66] distinguishing between MIMD (multiple instruction multiple data) and SIMD (single instruction multiple date) machines. In the MIMD category we distinguish two main concepts:

- shared memory systems and
- distributed memory systems[2].

In a shared memory architecture the processors communicate with each other via a common memory. In such *multi-processor systems* [Ensl 80] the interconnection network should guarantee that each processor has access to every memory module. The interconnection network should further guarantee, that there will be no conflicts among the processing elements, which want to access memory (nonblocking networks).

In distributed memory architectures each processing element has its own local memory. In such *multi-computer system* [Atha 88] the interconnection network should provide a (not necessarily direct) connection between every pair of processing elements (compare figure 1.1).

The network topology, defined as the abstract representation of the connections in the network [Hein 89], is a key factor in determing a suitable architectural structure. We distinguish between two types of topologies [Feng 81]:

- dynamic topologies, where the connections are established time dependently, either via common busses or via a switching network and
- static topologies, where there are dedicated links between the processing elements.

In this work we will review static topologies, mainly proposed for message passing distributed memory systems. We will explicitely rule out dynamic topologies of our discussion for two reasons:

[1] In [Ande 75], [Hwan 84], [Hobb 88] or [Trel 87, Trel 88] surveys of parallel computers can be found.

[2] A comparison between shared and distributed memory systems can be found for instance in [Bail 88]

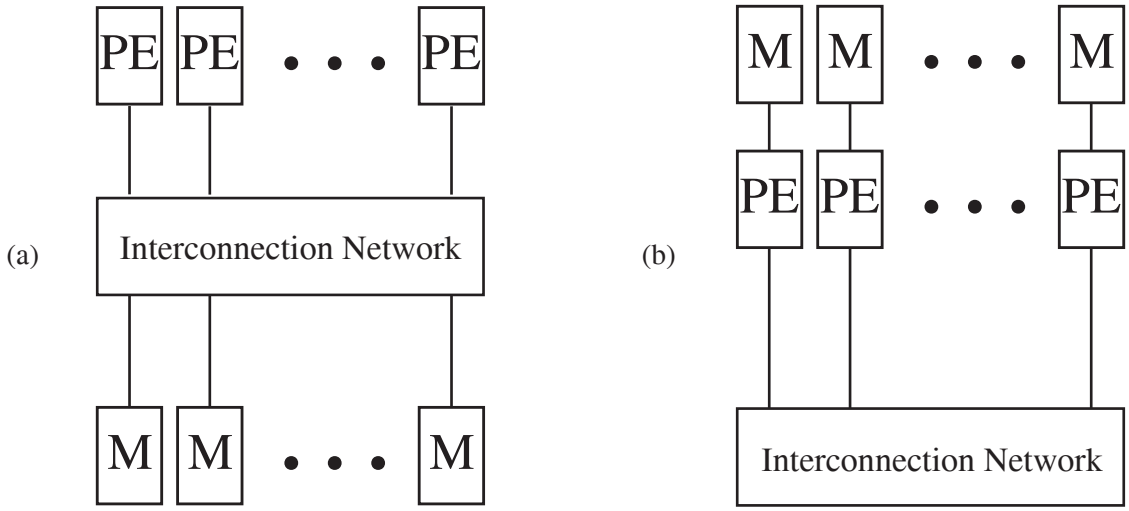


Figure 1.1: (a) Multiprocessor System (b) Multicomputer System

First, the comparison of both, static and dynamic topologies (which are mainly used in shared memory systems to interconnect processing and memory modules), is difficult, because of the different requirements posed on the network in shared and distributed memory systems. Second, there is already a lot of literature surveying and comparing dynamic topologies [3].

This work consists of two main parts: the first part (Chapters 2, 3 and 4) deals with a presentation and analysis of static topologies. We will start with establishing criteria for interconnection topologies (Section 2), which shall serve as a basis for the presentation and comparison of the topologies. Those criteria include aspects of *communication delays*, *connection costs*, *reliability* and *message passing*, in this work discussed under the term *routing*. In the next chapter (3) we will present topologies, which have been proposed in the literature for interconnection networks. I am aware of the impracticability to present all topologies, that have ever been suggested, so I want to focus in this work on the construction principles for the most important topologies, to give the reader a general idea of the variety in this research area. Extensive references will guide him to specific details and properties of topologies if desired. In Chapter 4 we will rate the topologies presented in the previous section according to some criteria of Chapter 2.

In the second part of my work I will discuss two special issues, namely the problem of finding topologies with minimum communication delays (Chapter 5) and the problem of routing in static networks (Chpater 6). In Chapter 5 I will refer to developments in this research field, and propose *Extended Chordal Rings* as a good solution to this problem. When dealing with aspects of routing, I will consider several special communication demands (i.e. point-to-point routing, broadcasting and gossiping) and analyse the requirements posed on the topology and the routing algorithm in order to support efficient (fast) routing. Besides this general considerations, I will also develop routing algorithms for Extended Chordal Rings.

[3] The interested reader might be referred to [Agra 78], [Adam 82], [Agra 83], [Bhuy 83], [Wu 84], [Hwan 84], [Ragh 86], [Gilo 86], or [Sche 91], which deal all with switching networks; topologies for bus oriented networks can be found in [Agra 86],[Fink 80], or [Fink 81].

Chapter 2

Network Characteristics

"est modus in rebus
sunt certi denique fines"
Horaz

A convenient model for the interconnection topology of multicomputers is a graph $G = (V, E)$ where V is the set of nodes representing the processing elements (PE) in the network and E is the set of edges representing links connecting processing elements. A network with bidirectional links is modelled by an undirected graph, and an edge between nodes x and y $x, y \in V$ will be denoted by $e = (x, y) \in E$. If the links in the multi-computer system are unidirectional we will use a directed graph, an arc from node x to node y is shown as $(x \mapsto y)$.

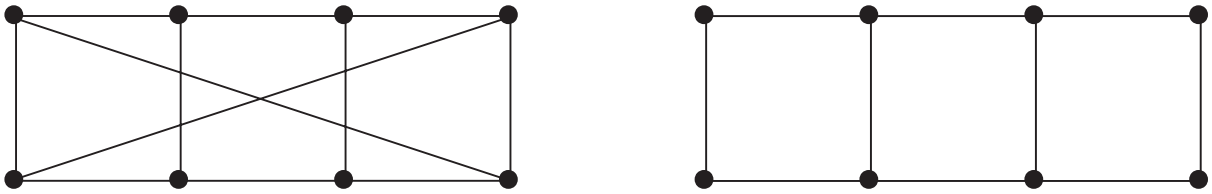
Based on this model interconnection network characteristics can be found by interpreting properties of graphs[1]. We will call this type of characteristics *static measures* since the network is investigated from a static point of view (for example no aspects of routing behaviour are considered). Those measures are used for comparing the size of a network, the connection costs (measured in the number of links), symmetry properties, but also as a first rough estimation of communication delays expressed in terms of the number of links between two PE's.

An important property for multi-computer systems are short communication delays. We need therefore *dynamic measures*, which will help us in analysing the network's communication behaviour. We are interested in short communication paths between processing elements (average distance), and in equally utilization of the PEs and links when routing a message in order to avoid congestures. The corresponding measures are visit ratio and worst through routing load.

The communication delays do not only depend on the number of links between PEs, but also on the effort for calculating the path between any two nodes in the network. We will discuss the problem of sending a message from one node to another under the term *routing*, and establish several criteria, that make up a "good" routing algorithm.

Another important feature in the design of parallel computers is the *reliability* and availability of the system components. We will present *deterministic* and *probabilistic* measures based on different criteria for acceptable network operation. We can either define a network to be operable, if there exists a path between any pair of processing elements in spite of faulty components (connectivity), or, more restrictive, if the failure of links and nodes will not lead to a large increase in communication delays (persistance). When considering probabilistic criteria, we will assign a certain failure probability to each system component (PEs and links), and determine the probability, that the whole network is reliable (again measured in the existance of a path between any two nodes). We will see that the calculation of those probabilistic criteria is much more difficult than the analysis with deterministic criteria.

[1] In [Hara 72] or [Berg 62] a good survey of graph invariants (properties) can be found.



(a) regular topology of degree 3 (b) irregular topology

Figure 2.1: Regularity in graphs

Extensibility, i. e. the ability to increase the system capability, is a further aspect in the design of a parallel system. In instances, where a specific design has to be configured for a variety of applications, it is often desirded to change the number of system components (processing elements) according to the computational requirements of the particular problem. It should be possible to increase the number of system components in an existing topolgy without beeing forced to rearrange the whole network.

2.1 Static Measures

Node number N The node number is the cardinality of the node set V or the number of processing elements (PE's) in the network, also called *size* of the network or *order* of a graph.

Definition 1 *The node number, denoted by N, is defined as*

$$N = |V|$$

where V denotes the set of nodes.

The number of PE's in a multicomputer system is sometimes used as a classification scheme distinguishing between small ($N < 10$), medium ($N < 1000$) and large systems ($N > 1000$) [Seit 84].

Degree d The degree is the maximum number of edges incident to any node. It represents the maximum number of links or communication channels per PE.

Definition 2 *The degree of a node node degree, denoted by d_i, is the number of nodes adjacent to node i.*

The maximum of all node degrees is the degree of the graph denoted by d

$$d = \max_i\{d_i\}$$

A topology is called *regular* if every node has the same node degree otherwise it is called *irregular* (see figure 2.1).

If we neglect the case of disconnected graphs or graphs with multiple edges, then there are two extremal values for d yielding two extremal topologies. In the first case d equals $(N-1)$ so each node is connected to all others. This topology is a complete graph on N nodes, denoted by $K(N)$. The other extreme is to have only $d = 2$ edges per node connecting it with its two neighbours. The resulting structure is a ring $R(N)$ of N nodes.

Since the degree is usually bounded by hardware restrictions[2] it is often desirable and necessary that the degree does not grow when extending the size of a network.

Total Number of Links L The total number of links is the cardinality of the set of edges E. It is a measure for connection costs.

Definition 3 *The total number of links, denoted by L, is defined as*

$$L = \begin{cases} (\sum_{i=0}^{N-1} d_i)/2 & \textit{in irregular topologies} \\ Nd/2 & \textit{in regular topologies} \end{cases}$$

The regular graph in figure 2.1 has $L = 12$, whereas the irregular graph has $L = 10$, and therefore lower connection costs.

In order to keep connection costs as small as possible, the total number of links should only grow linearly with the total number of PE's (which corresponds in regular topologies to a constant degree). In the worst case L grows quadratically with N (completely connected networks).

Connectedness $\mathcal{C}$ [Fers 91a] The connectedness is a measure for strength of coupling among the processing elements, comparing the number of edges actually in the network to the number of edges that would be necessary to establish a complete network $K(N)$.

Definition 4 *The connectedness, denoted by $\mathcal{C}$, is defined as*

$$\mathcal{C} = L/L_{K(N)}$$

where L denotes the number of edges in the network and $L_{K(N)}$ denotes the number of edges in a complete network of the same order.

If the connectedness is smaller than $2/N$ the graph must be disconnected, if it is larger than 1 there exists at least one multiple edge between one pair of nodes[3]. A small value of $\mathcal{C}$ indicates that there are relatively few links in the network, i. e. the processing elements are loosely coupled, a value close to one corresponds with high connection costs.

Symmetry [Fost 32] A desirable feature for interconnection networks is symmetry, since a high degree of symmetry in general simplifies routing and leads to general purpose networks. The optimum is a completely symmetric structure where the network looks exactly the same viewed from an arbitrary node.

Definition 5 *Two nodes x and y in a graph G are similar, if for some automporphism[4] α of G, $\alpha(x) = y$ with $x, y \in V$.*

Two edges (x_1, y_1) and (x_2, y_2) in a graph G are similar, if for some automporphism α of G, $\alpha((x_1, y_1)) = (x_2, y_2)$ with $(x_1, y_1), (x_2, y_2) \in E$.

A graph is called node-symmetric, if every pair of nodes is similar.

A graph is called edge-symmetric, if every pair of edges is similar.

A graph is called symmetric, if it is node- and edge-symmetric.

[2] For instance the transputer has $d = 4$.

[3] Those bounds are obtained by using the line as structure with the smallest possible number of edges for a connected graph ($L = N - 1$), and the complete network ($L = N(N-1)$) as the other extreme.

[4] An automorphism is a 1-1 mapping of a graph onto itself (permutation).

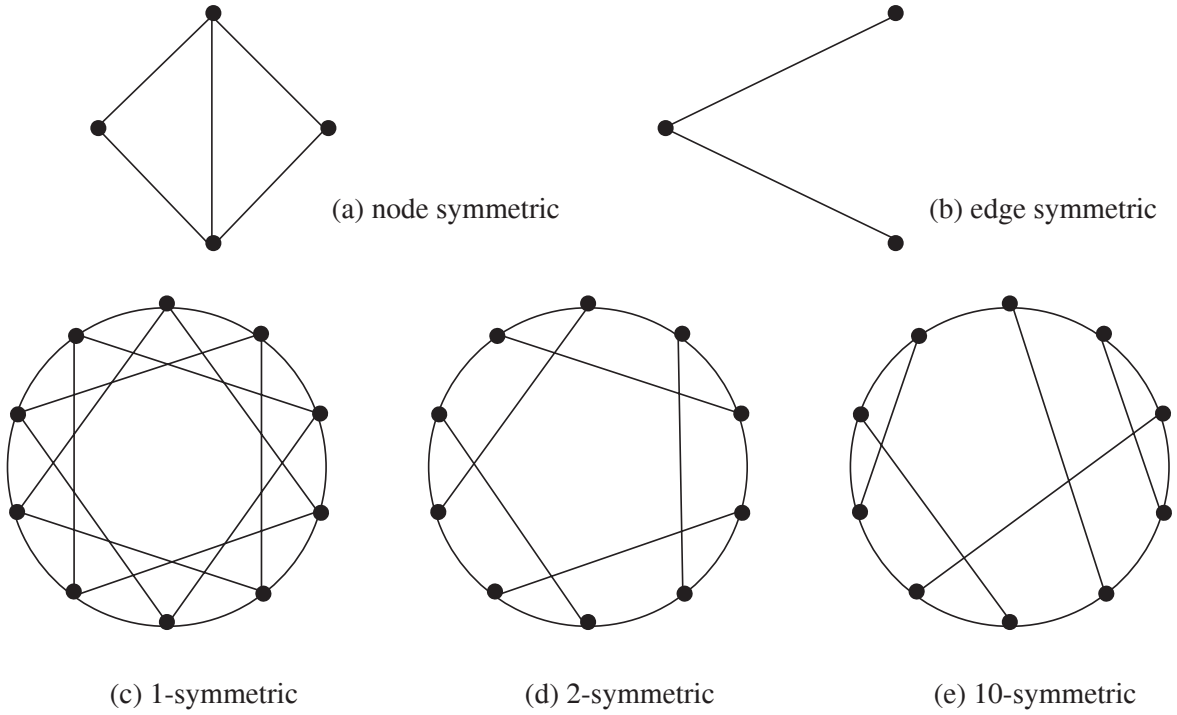


Figure 2.2: Symmetry in graphs

Figure 2.2 shows a node-symmetric graph which is not edge-symmetric (a) and an edge-symmetric graph which is not node-symmetric (b).

Defintion 5 only allows us to say whether a graph is (node-,edge-) symmetric or not. But it would be useful to quantify the degree of symmetry, i. e. to distinguish between more or less not symmetric graphs. We will therefore give the following definition:

Definition 6 *A graph $G(V, E)$ is said to be s-symmetric if the number of nodes can be divided into s subsets $V_1, V_2, \ldots, V_s$ with $\bigcup_{i=1}^{s} V_i = V$ and $\bigcap_{i=1}^{s} V_i = 0$ and in each subset V_i every pair of nodes must be similar (analogous for edges).*

The graphs (a) and (b) in figure 2.2 are 2-symmetric. A 1-symmetric graph is equivalent to a symmetric graph (see figure 2.2 (c)), (d) shows a 2-symmetric structure, and an N-symmetric graph (also called *asymmetric*) is shown in figure 2.2 (e).

Diameter k The diameter is a worst case measure for the length of a path between any two nodes in the network, measured in the number of edges that have to be passed going from one node to another. The diameter is also called *maximum internode distance* [Span 82].

Definition 7 *The diameter, denoted by k, is defined as*

$$k = \max_{i,j}\{k_{i,j}\}$$

where $k_{i,j}$ denotes the distance between node i and node j.

In message passing multicomputer systems, the communication paths should be short, so the interconnection network should have a small diameter. The optimum topologies with respect to diameter are complete networks ($k = 1$), but they have prohibetively high connection costs (a degree of $N-1$ and a total number of $N(N-1)$ links). If we look at the line as a structure with minimum connection costs, we have a diameter of $N-1$ (the length of the path from the first node to the last) and one might conjecture, that there is a trade off between degree and diameter, i. e. a higher degree corresponds to a lower diameter and vice versa. In fact, such a relation does exist and was found by E. F. Moore. The so called *Moore Bound* establishes an upper bound for the total number of nodes under given degree and diameter[5].

$$N_{max} = \begin{cases} 2k+1 & \text{if } d = 2 \\ \frac{d(d-1)^k - 2}{d-2} & \text{if } d > 2 \end{cases} \tag{2.1}$$

and resolving this equation with respect to k yields a lower bound for the diameter in a network with given N and d:

$$k_{min} = \begin{cases} \frac{(N-1)}{2} & \text{if } d = 2 \\ \frac{\log_{d-1}(N(d-2)+2)}{d} & \text{if } d > 2 \end{cases} \tag{2.2}$$

So our conjecture has turned out to be true, there is a tradeoff between degree and diameter and it is not possible to design a network which minimizes both, d and k for a given number of nodes. Usually d is the given parameter and the problem is either to maximize the number of nodes for a given degree and diameter (see Section 5.1) or to find a network with minimum diameter for given N (see Section 5.2).

If we want to compare the diameter of graphs with different degrees, it seems not fair to look merely at diameter, because it is easy to obtain a low diameter when allowing a higher degree. So it would be better to use for comparison either the product of the degree by the diameter[6] dk, or the deviation of the diameter from the minimum diameter ($k - k_{min}$) [Cont 83]. Let us consider the graphs in figure 2.2 (c) and (d) as an example: both are of diameter 3, but the first one has degree 4, whereas the other one has degree 3, so the second graph is better, when using dk for comparison. If we calculate the minimum diameter for $N = 10$ and $d = 4$ ($k_{min}(10, 4) = 2$) and for $d = 3$ ($k_{min}(10, 3) = 2$) we can see that both graphs differ by 1 from the minimum diameter, so none of them is optimum.

2.2 Dynamic Measures

This sections deals with measures that characterize the network's communication behaviour. We are interested in communication delays (average distance) and in utilization of links and processing elements. Dynamic measures depend on the communication requests among the PE's. Besides from special requests resulting from a specific program structure we can distinguish between the following distributions [Reed 87b, Reed 87a]:

- Uniform message routing, where the probability that a PE wants to send a message to another PE is the same for all pairs of PE's.
- sphere of locality message routing, i. e. every PE has a small number of PE's with whom it communicates at high probability, whereas communication to PE's outside this sphere of locality is less probable.

[5] In Appendix A.1 the derivation of the Moore Bound is shown.

[6] This measure is sometimes called *cost factor* ξ [Bhuy 84].

- decreasing probability message routing, i. e. the probability for communication requests between PE's decreases with their distance.

We will restrict our investigations to the case of uniform message routing distribution for two reasons: on the one hand, we are interested in general purpose architectures so it seems reasonable to assume uniform distribution, and on the other hand dynamic measures are easier to calculate if equally distributed communication requests are assumed. However, there are some topologies which fit well to other routing distributions, and we will mention them when discussing the topologies (section 3).

Average Distance μ [Prad 82] The average distance is the number of links which have to be crossed on average for communication between any two processing elements.

So the value for average distance depends on the frequency at which PE's whish to communicate with each other (as mentioned above uniform distribution will be assumed), and on the routing algorithm, where we will assume that the shortest path between any PE's will always be chosen.

Definition 8 *The average (internode) distance*[7]*, denoted by μ, is defined as*

$$\mu = (\sum_{i=1}^{k} iP_i)/(N^2 - N)$$

where P_i denotes the number of shortest paths of lengths i between all pairs of distinct nodes.

If we consider again the graphs of figure 2.2 (c), (d) and (e) we can see that the first graph has an average distance of $\mu = 1.667$, the two others have $\mu = 1.889$, so the communication delays in the first graph will be shorter on average although all structures have the same communication delays in worst case ($k = 3$).

As for the diameter it is also possible to derive a lower bound for the average distance from the Moore Bound[8]:

$$\mu_{min} = \frac{\sum_{i=1}^{k_{min}} i\, d\,(d-1)^{i-1} \; - k_{min}(N_{max} - N)}{N - 1} \tag{2.3}$$

If the topology is symmetric it is sufficient to calculate the average distance for a single node according to the following equation yielding the same result as in definition 8:

$$\mu_j = \frac{\sum_{i=1}^{k} iN_i}{N - 1}$$

where N_i denotes the number of nodes at distance i from node j [Agra 86]. If the investigated topology is only s-symmetric (see definition 6) the calculation of the average distance will yield different results depending on the starting node j. So μ_j has to be calculated for one node in every symmetry class[9] and then

$$\mu = \frac{\sum_{j=1}^{s} \mu_j S_j}{s}$$

[7] Prior [Prio 90] gives a slightly different definition for the average distance, called mean internode distance $mid = (\sum_{i=1}^{k} iP_i)/N^2$. The mean internode distance will always be smaller then the average distance, because mid also counts the paths from a node to itself with length 0 which increases the denominator.

[8] The derivation is shown in Appendix A.4.

[9] Since the effort for calculation grows with s (the number of symmetry subsets), one might be satisfied with an estimation of μ based on a subset of starting nodes for calculation.

will give the same result as definition 8, where S_j denotes the number of nodes in symmetry class j.

In Appendix A.2 the calculation of average distance is shown for some structures by example.

Normalized Average Distance μ_{norm} [Agra 86],[Raab 88] A problem with the average distance arises in comparing topologies with different degree. A higher degree would cause the average distance to decrease and the topology with higher degree would be estimated better than the one with lower degree. The normalized average distance tries to eliminate this incomparability.

Definition 9 *The normalized average distance, denoted by μ_norm, is defined as*

$$\mu_{norm} = \mu d$$

where μ denotes the average distance of a topology and d denotes the degree.

When comparing the graphs of figure 2.2 (c) and (d), we should better use the normalized average distance. The first graph (c) is of degree 4 and has $\mu_{norm} = 6.668$, the other (d) is of degree 3 and has $\mu_{norm} = 5.667$, the comparison based on μ has shown an opposite result.

Visit ratio V [Reed 87b],[Prio 90] The visit ratio is a measure for the average utilization of a link, quantifying how often a single link will be used on average when routing a message from one PE to another PE.

Definition 10 *The visit ratio, denoted by V, is defined as*

$$V = \mu / L$$

where μ denotes the average distance and L denotes the total number of edges.

A visit ratio value close to 1 indicates that a message will have to pass nearly all links when beeing routed from one PE to another. A lower bound for V is obviously given by $1/L$.

The graph in figure 2.2 (c) has $V = 0.083$ (which exceeds the lower bound by 66 %) and (d) and (e) have $V = 0.126$ (which exceeds the lower bound by 89 %). Take for a comparison the visit ratio in a binary tree of 7 nodes, which has $V = 0.38$ (which exceeds the lower bound by 228 %!).

A low visit ratio indicates a low probability for the occurance of additional delays due to "congested" links when routing many messages concurrently in the network and vice versa. So we see that the "traffic" in a binary tree is comparatively high but we do not now whether some links are more often used than others. In order to analyse the load distribution we will define worst case measures in the following.

Worst Through Routing Load γ [Prio 90] The worst through routing load is defined as the largest number of routes[10] that pass through any node (respectively edge[11]). So it is also a measure for utilization of PE's and links but in contrast to the visit ratio it determines the visits to each link seperately.

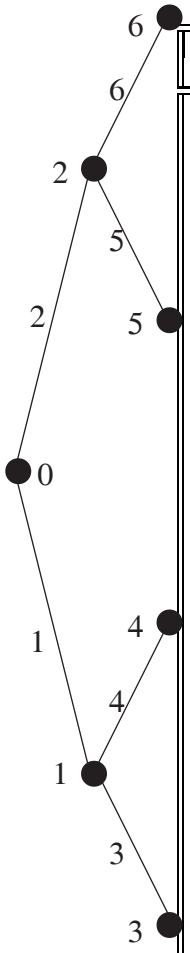
Definition 11 *The worst through routing load for nodes, denoted by γ^{node}, is defined as*

$$\gamma^{node} = \max_i \{\gamma_i^{node}\}$$

where γ_i^{node} denotes the number of routes passing through node i.

[10] A route is a path from one node to another node where no edges or nodes will be visited twice.

[11] The worst through routing load for links is also called most-loaded link load [Prio 90].



routes	nodes 0	1	2	edges 1	2	3	4	5	6	γ^{node} $= \max_i \gamma_i^{node}$	γ^{edge} $= \max \gamma_i^{edge}$
0,1				1							
0,2					1						
0,1,3		1		1		1					
0,1,4		1		1			1				
0,2,5			1		1			1			
0,2,6			1		1				1		
1,0				1							
1,0,2	1			1	1						
1,3						1					
1,4							1				
1,0,2,5	1		1	1	1			1			
1,0,2,6	1		1	1	1				1		
2,0					1						
2,0,1	1			1	1						
2,0,1,3	1	1		1	1	1					
2,0,1,4	1	1		1	1		1				
2,5								1			
2,6									1		
3,1,0		1		1		1					
3,1						1					
3,1,0,2	1	1		1	1	1					
3,1,4		1				1	1				
3,1,0,2,5	1	1	1	1	1	1		1			
3,1,0,2,6	1	1	1	1	1	1			1		
4,1,0		1		1			1				
4,1							1				
4,1,0,2	1	1		1	1		1				
4,1,3		1				1	1				
4,1,0,2,5	1	1	1	1	1		1	1			
4,1,0,2,6	1	1	1	1	1		1	1	1		
5,2,0			1		1			1			
5,2,0,1	1		1	1	1			1			
5,2								1			
5,2,0,1,3	1	1	1	1	1	1		1			
5,2,0,1,4	1	1	1	1	1		1	1			
5,2,6			1					1	1		
6,2,0			1		1				1		
6,2,0,1	1		1	1	1				1		
6,2									1		
6,2,0,1,3	1	1	1	1	1	1			1		
6,2,0,1,4	1	1	1	1	1		1		1		
6,2,5			1					1	1		
γ_i^{node}	18	18	18							18	
γ_i^{edge}				24	24	12	12	12	12		24

Table 2.1: Worst through routing loads in a binary tree

	type 1	type 2	type 3	type 4	type 5	type 6	type 7	type 8
1.	yes	yes	yes	yes	no	no	no	no
2.	yes	yes	no	no	yes	yes	no	no
3.	yes	no	yes	no	yes	no	yes	no

Table 2.2: Routing types

Definition 12 *The worst through routing load for edges, denoted by γ^{edge}, is defined as*

$$\gamma^{edge} = \max_i \{\gamma_i^{edge}\}$$

where γ_i^{edge} denotes the number of routes passing over edge i.

The total number of routes in the network is given by $N(N-1)$ (one route for each pair of distinct nodes) and we can establishe an upper bound for the worst through routing load $\gamma < N(N-1)$. It would be a desirable feature of an interconnection network that the communication load is equally distributed among PE's so

$$\gamma_{opt}^{node} = \frac{N(N-1)}{N} = (N-1)$$

and analogously for links

$$\gamma_{opt}^{edge} = \frac{N(N-1)}{L}$$

Table 2.1 shows the worst through routing loads for nodes and edges in a binary tree of 7 nodes, labelled from 0 to 6 (as shown beside the table). The first column of this table lists all routes in the tree as a sequence of the node labels, the next three columns indicate whether node i is used in this route for transmission (we can omit the leaf nodes since they will never have to pass on any messages). A 1 in position (i, j) denotes, that the node (edge) in column j lies on the route in row i. In the last two rows the routing load for every node γ_i^{node} and edge γ_i^{edge} is calculated (by adding up the ones in the column) and the maxima in these rows determine the worst through routing loads, shown in the last two columns.

2.3 Efficient Routing

For processing elements in multicomputer systems that communicate via message passing an efficient routing strategy is an essential necessity. In Section 6 we will discuss different types of routing requirements (one-to-one routing, broadcasting, gossiping) and analyse the requirements for networks and appropriate routing functions.

For the survey and comparison of topologies (sections 3 and 4 we will concentrate on simple one-to-one routing, i. e. one (arbitrarily chosen) PE wants to send a message to another (arbitrarily chosen) PE.

In order to support fast one-to-one communication the network must have small values for diameter and average distance and the determination of a path between two processing elements should be done locally with minor computational efforts and memory requirements (the PE's should not have to store an allocation table containing the whole network). The routing algorithm should also be applicable in the case of PE or link failures.

We define the desirable properties of a routing algorithm for one-to-one communication as follows:

1. It is not necessary to explicitely store the whole topology in every node.

2. The shortest path between any two PE's is taken.

3. If there exists a path between any two nodes this path is found despite of the failure of PE's or links.

The first property is fulfilled optimum by computational routing algorithms, where the path between two nodes is found by comparison of the source and destination addresses.

We can classify a routing algorithm by specifiying which properties are fulfilled or not and define the "types of routing" as shown in table 2.2 (the numbers in the leftmost column refer to the properties stated above and "yes" or "no" indicates whether this property is fulfilled or not).

When discussing the topologies Chapter 3 we will not always give a detailed description of the routing algorithm but rather specify the routing type and refer the interested reader to the corresponding literature.

2.4 Reliability

A reliable network is able to remain functioning in spite on failures in links or processors. That means that it is still possible for the remaining (non faulty) PE's to communicate with each other. In order to do this, there must be mechanisms for detection and location of failures (*fault diagnosis* [Arms 81], [Bhat 82]) and the possibility to bypass faulty components (*fault tolerance*).

The ability of fault tolerance requires redundant paths between pairs of PE's. The worst case is that the network becomes disconnected because of link or PE failures. The first measure that we will present is based on graph-theoretic ideas of *connectivity* defined as the minimum number of nodes whose removal seperates the graph into at least two components. Some attempts have been made to generalize this ideas (*generalized connectivity* and *persistence*) [Boes 70], [Skil 89]. In all those *deterministic measures* we only consider the worst case. Those measures are sometimes called *vulnerability* measures ([Fran 71], [Seng 87], [Berm 83]), because they show, how easy or difficult it is, to destroy (disconnect) the topology.

In a *probabilistic approach* every link and PE is assigned a certain failure probability and the goal of the analysis is to determine the probability that the whole network is operational, i. e. not disconnected [Tain 76]. The analysis is quite simple if the network can be represented as a series-parallel graph, otherwise more sophisticated analysis methods are necessary. In [Hwan 81] a survey of the literature dealing with reliability analysis techniques can be found. Those techniques are very time consuming (they usually belong to the class of NP-hard problems) and are thus not feasible for large networks. So it is necessary to make assumptions which simplify the analysis and to define criteria based on approximative calculations.

2.4.1 Deterministic Measures

Connectivity and Cohesion λ,κ [Boes 70] Connectivity and cohesion measure the "difficulties for disrupting" the system by removing nodes and edges.

Definition 13 *The node-connectivity, denoted by κ, is the minimum number of nodes whose removal results in a disconnected or trivial*[12] *graph, where removing a node means to remove the node itself and all edges adjacent to it.*

A graph is called n-connected if $\kappa = n$.

Definition 14 *The edge-connectivity or cohesion, denoted by λ, is the minimum number of edges whose removal results in a disconnected or trivial graph.*

A graph is called n-edge-connected if $\lambda = n$.

[12] A trivial graph has exactly one node

By definition, the connectivity of a graph will never be larger than its cohesion. Clearly the cohesion is bounded by the minimum node degree in the graph which is in turn bounded by the average node degree[13] and so the following inequality holds (see [Whit 32] for a proof):

$$\kappa \leq \lambda \leq d_{min} \leq \begin{cases} 0 & \text{for } L < N-1 \\ \lfloor 2L/N \rfloor & \text{for } L \geq N-1 \end{cases}$$

A graph with $\lambda = 2L/N$ is optimum and is called *super-λ* [Huan 89].

Whitney [Whit 32] discovered and proved the following criterion for connectivity:

Theorem 1 *A graph is n-connected if and only if there are at least n node disjoint[14] paths between any pair of nodes.*

The analogous theorem for edge-connectivity could not be proven until 30 years after by Ford and Fulkerson [Ford 62]:

Theorem 2 *A graph is n-edge-connected if and only if there are at least n edge disjoint[15] paths between any pair of nodes.*

These two theorems are important for analysing the connectivity of networks. If we assign a weight of value 1 to every edge in the network the value of λ is equivalent to the minimum over all minimum cuts between any two nodes in the network. So the problem is to find the minimum cuts for all pairs of nodes which can be done by the algorithm of Ford an Fulkerson [Ford 62]. The determination of κ is analogous after transforming the network (splitting the nodes resulting in a network with $2N$ nodes and $L+N$ edges). Frisch [Fris 67],[Fris 69] has developed an algorithm which allows the calculation of κ without transformation. Several other algorithms can be found in [Gomo 61],[Klei 69] and [Frec 70].

Connectivity and cohesion only give a very first view of the networks failure behaviour. They only tell us, whether the system will stand the failure of a certain number of nodes (respectively links) or not. We know neither, what will happen, if less failures occur (will this leed to an increase in diameter?), nor, how harmful the failure of κ nodes (or λ links) is (will only be a single node seperated or nearly half of the system?). Inspite of these drawbacks, those measures are often used for comparing the fault tolerance of networks, because algorithms for their calculation are known (see the previous paragraph).

Generalized Connectivity and Cohesion $\lambda(x)$,$\kappa(x)$ [Boes 70] More general measures of network reliability based on cutsets are generalized connectivity and generalized cohesion. They measure the number of nodes (respectively edges) whose removal separates a graph with x nodes from the original graph.

Definition 15 *Let x be an integer with $1 \leq x \leq \lfloor N/2 \rfloor$.*

The Generalized connectivity, denoted by $\kappa(x)$, is defined as the minimum number of nodes whose removal seperates a subgraph of order less than or equal x and all remaining parts of the graph are of order not less than x.

The Generalized cohesion denoted by $\lambda(x)$ is defined as the minimum number of edges whose removal seperates a subgraph of order less than or equal x and all remaining parts of the graph are of order not less than x.

[13] The average node degree is the number of edges that a node has on average, calculated by dividing the total number of edges by the total number of nodes L/N. Since each edge is incident to two nodes, we must double L/N in order to obtain the average node degree.

[14] Two paths are said to be node disjoint, if there exists no node which occurs in both paths.

[15] Two paths are said to be edge disjoint, if there exists no edge which occurs in both paths.

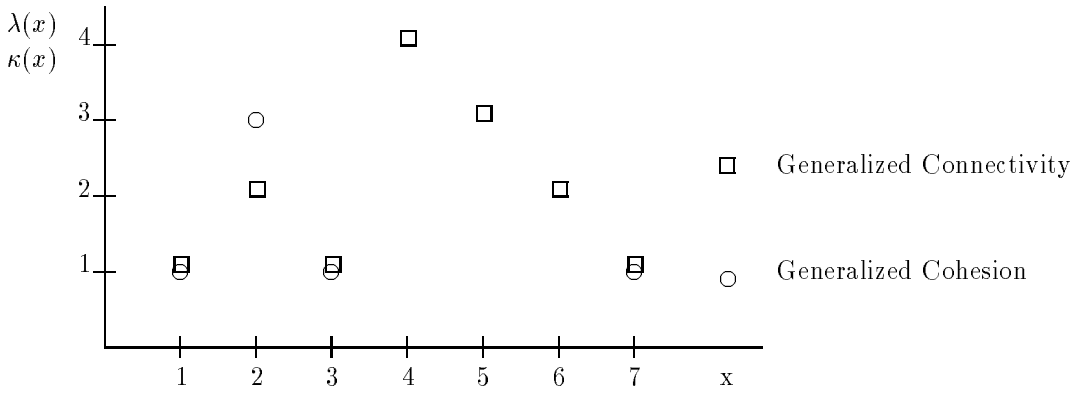


Figure 2.3: Generalized Connectivity and cohesion in a Binary tree with 15 nodes

It follows from the definition that x must be in a range of $[1, \frac{N}{2}]$.

Generalized connectivity and cohesion allow to discriminate between the isolation of a single node and the separation of larger parts of the network and are helpful in detecting parts of the graph that are not tightly connected to the remainder. So generalized connectivity[16] and generalized cohesion[17] allow a more detailed look at the network than connectivity and cohesion. If we have determined $\kappa(x)$, it is easy to find κ, because κ is defined as the minimum number of nodes whose removal seperates the graph (into components of arbitrary order) and the generalized connectivity shows the minimum number of nodes required for separating components of all possible orders, and therefore $\kappa = min_x\{\kappa(x)\}$. The same relation is true for cohesion and generalized cohesion.

The disadvantage of these measures is a higher effort for calculation, no explicit algorithms are known.

Figure 2.3 shows the generalized connectivity and cohesion for a binary tree on 15 nodes. It is always possible to seperate a complete subtree by deleting a single node or edge ($\lambda(2^i - 1) = \kappa(2^i - 1) = 1$). A subgraph of size 2 is obtained by deleting 2 nodes (1 and 7, for example) or 3 edges (2,3 and 4). Note that it is not possible to seperate subgraphs of sizes 4,5 and 6 by deleting edges because the remaining parts of the tree would be smaller.

Persistence and Edge-Persistence ρ Wilkov [Wilk 72] and Bollobás [Boll 78] have investigated the minimum number of nodes (and edges) whose removal causes diameter to increase up to a certain bound k'. We want to follow the terms and definitions from Boesch [Boes 81] and Exoo [Exoo 82] where $k' = k + 1$, an increase in the diameter of the graph by one.

Definition 16 *The persistence, denoted by ρ_{node}, is the minimum number of nodes whose removal cause an increase in diameter.*

Definition 17 *The edge-persistence, denoted by ρ_{edge}, is the minimum number of edges whose removal causes an increase in diameter.*

It is obvious that $\rho_{node} \leq \kappa$ and $\rho_{edge} \leq \lambda$ and one might conclude that ρ_{node} (ρ_{edge}) equals the minimum of the maximum number of node-(edge-)disjoint paths with length not greater than

[16] called *connectivity function* $c(x)$ in [Skil 89]

[17] al so called minimum m-degree $\delta(m)$ in [Boes 70]

k between every pair of nodes. This assertion was made and proven by [Boes 81], but Exoo [Exoo 82] has shown that this is only true for graphs with diameter $2 \leq k \leq 4$ for persistence and $2 \leq k \leq 3$ for edge-persistence.

Persistence and edge persistence seem to be the most useful deterministic measures, because an increase in diameter corresponds to an increase in communication delays. If the communication delays become too large, it might be better (faster) to use traditional computers instead of parallel systems.

2.4.2 Probabilistic Measures

Reliability and Unreliability Function $\mathcal{R}$,$\mathcal{U}$ If we want to determine the probability of a graph beeing connected (disconnected) under the assumptions that all nodes are perfectly reliable (i. e. their failure probability is 0) and all edge failures occur independently with the same probability, we can calculate the reliability function of the graph [Kelm 67].

Definition 18 *The reliability function, denoted by $\mathcal{R}$, is defined as*

$$\mathcal{R} = \sum_{i=N-1}^{L} S_i (1-p)^i p^{L-i}$$

and the unreliability function, denoted by $\mathcal{U}$, is defined as

$$\mathcal{U} = \sum_{i=\lambda}^{L} D_i p^i (1-p)^{L-i}$$

where p denotes the failure probability of a single edge, S_i denotes the number of spanning subgraphs[18] containing i edges and D_i denotes the number of disconnected spanning subgraphs[19] containing $(L-i)$ edges.

The formula for reliability is obtained from the following considerations: We can calculate the probability that exactly i edges are non faulty (and consequently that $L-i$ are faulty) by $(1-p)^i p^{L-i}$ (since the edge failures are assumed to occur independently). We know, that more than $N-1$ faulty nodes will always result in a disconnected graph, therefore at least $N-1$ edges must be non faulty and at most all L links can be non faulty. Since there are in general several possibilities for removing $L-i$ edges without disconnecting the graph, we need the number of all possible spanning subgraphs containing i edges, S_i, for $i = N-1, N, \ldots, L$, and obtain a closed form for $\mathcal{R}$, similar to the binomial distribution, by weighting the probability for the occurance of i nonfaulty node with S_i. Similar considerations lead to $\mathcal{U}$. Note only, that it is sufficient to start with the removal of λ links, since the removal of less links would not disconnect the graph.

Since both, $\mathcal{R}$ and $\mathcal{U}$ are probabilities, their definition range is $0 \leq \mathcal{R} \leq 1$ and $0 \leq \mathcal{U} \leq 1$, and the following equation holds: $\mathcal{R} = 1 - \mathcal{U}$.

The calculation of $\mathcal{R}$ or $\mathcal{U}$ belongs to the group of NP-hard problems (because it is necessary to determine all spanning subtrees) but can be approximated [Prov 83]:

[18] A graph $G_{sub}(V_{sub}, E_{sub})$ is called a spanning subgraph of $G(V,E)$ if G_{sub} is connected and $V_{sub} = V$ and $E_{sub} \subset E$.

[19] A graph $G_{sub}(V_{sub}, E_{sub})$ is called a disconnected spanning subgraph of $G(V,E)$ if G_{sub} is disconnected and $V_{sub} = V$ and $E_{sub} \subset E$.

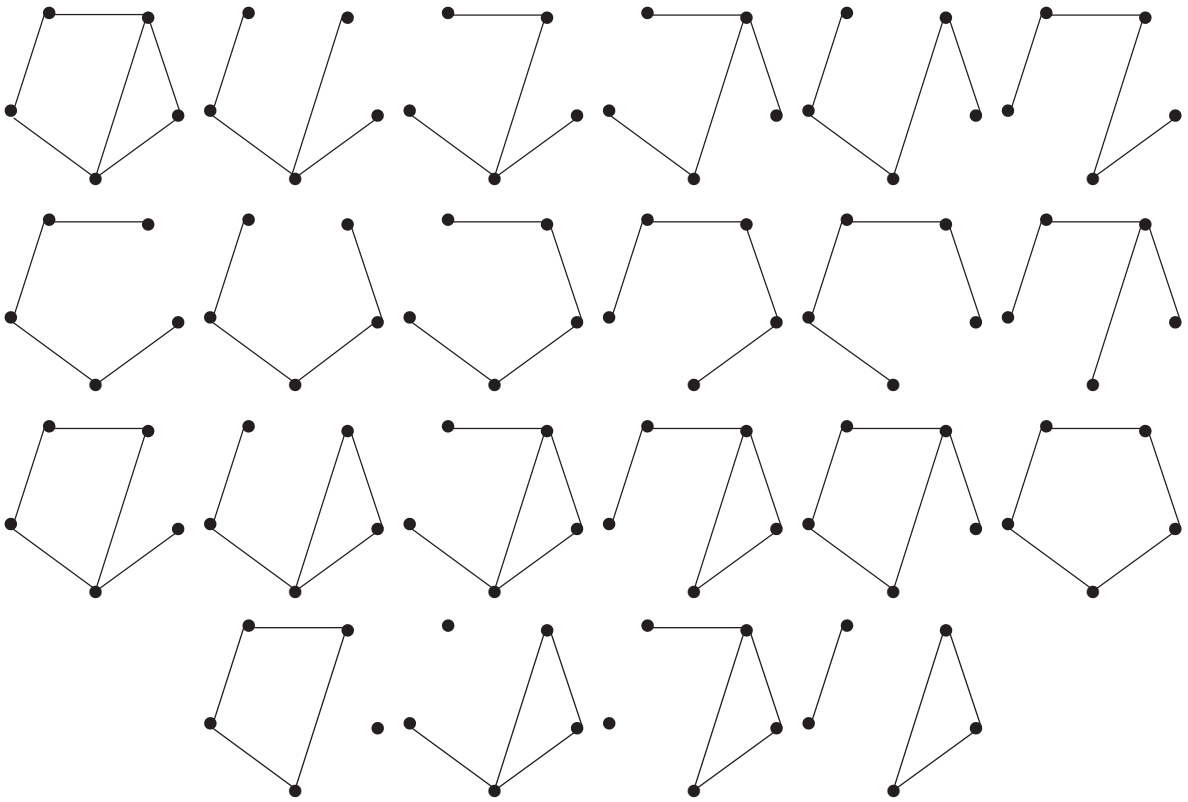


Figure 2.4: Spanning subtrees for reliability analysis

$$\mathcal{R} \approx \begin{cases} S_{N-1}(1-p)^{N-1} & \text{if } p \text{ is close to 1} \\ 1 - D_\lambda p^\lambda & \text{if } p \text{ is close to 0} \end{cases}$$

In the design of reliable multicomputer systems one is interested in maximizing the value of $\mathcal{R}$ and it has been shown by Bauer et al. [Baue 85] that this optimization problem is equivalent to

- finding a graph with a maximum number of spanning trees if p is large or
- finding a graph with maximum λ and a minimum number of disconnected spanning subgraphs (exactly $D_\lambda \rightarrow min$) if p is small.

Figure 2.4 shows all spanning subtrees for a small example graph (the left one in the first row) in the first three rows ($S_4 = 11$; $S_5 = 6$; $S_6 = 1$) and all disconnected spanning subtrees with 4 ($L - \lambda$) edges in the last row ($D_2 = 4$). In table 2.3 the exact and approximative values for $\mathcal{R}$ are shown for a low and a high failure probability.

Frank [Fran 69] has investigated network reliability under the opposite assumptions, i. e. all edges are perfectly reliable and the nodes are assigned to a certain failure probability q. Here

$$\mathcal{U} = \sum_{i=\kappa}^{N} N_i q^i (1-q)^{N-i}$$

	$p = 0.01$	$p = 0.95$
$\mathcal{R}$ exact	0.99596	0.00006384
$\mathcal{R}$ approx.	0.9996	0.00006875

Table 2.3: Reliability function

where N_i denotes the number of disconnected subgraphs (components) resulting from the removal of i nodes and κ is the connectivity of the graph. It is also possible to approximate $\mathcal{U}$ by the first term of the sum if q is sufficiently small, $\mathcal{U} \approx N_\kappa q^\kappa$.

When comparing topologies (sections 3 and 4) we will mainly concentrate on edge connectivity (in terms of the existence of redundant paths between any two nodes) as a measure of fault tolerance since the evaluation of all the other measures would be to expensive.

2.5 Extensibility

A desirable property, unfortunately possessed by only a few topologies, is ease in extensibility. Extending a topology means adding more nodes to the network, without changing the basic properties and characteristics of the topology. An example shall illustrate this: if we add a node via a new link to one of the two "end" nodes, the topology is still a line, and we have extended the number of nodes. If we add a node by connecting it to any other node in the line, the resulting structure is no longer a line, the basic properties are changed. A network is optimum extensible if it possesses the following properties:

1. An increase in the number of nodes does not cause an increase in the network degree.
2. It is possible to add nodes without the necessity of rearranging the existing structure.
3. It is possible to extend the structure to arbitrarily values of N.

The first property becomes most important when the degree is limited by hardware restrictions. If a graph possesses the second property it will be called *hierarchical*, if it possesses the last property then the graph is *extendible by increment 1* [Reed 87a].

We can classify the topologies according to these demands by defining the following categories (the numbers in the leftmost column refer to the properties stated above and "yes" or "no" indicates whether this property is fulfilled or not):

	type 1	type 2	type 3	type 4	type 5	type 6	type 7	type 8
1.	yes	yes	yes	yes	no	no	no	no
2.	yes	yes	no	no	yes	yes	no	no
3.	yes	no	yes	no	yes	no	yes	no

Table 2.4: Extensibility types

We will use this classification in the Chapter 3 when discussing the topologies.

Chapter 3

Survey of Topologies

"quis, quid, ubi, quibus auxiliis,
cur, quomodo, quando"
Sallust

A lot of topologies for static link oriented interconnection networks have been proposed in the literature. When a "survey" of topologies is given, usually only the "classical" structures, such as trees, rings or hypercubes, are mentioned [Witt 81, Dunc 90].

I have tried to give a systematic presentation helping the reader to obtain a general view of the variety of topologies. One classification proposed by Feng [Feng 81] seperates the topologies according to the dimension necessary for layout, or in terms of graph theory, whether the graph is planar (1- or 2-dimensional) or not (n-dimensional). This classification is not useful, since it is very difficult to decide whether a graph is planar or not [Hara 72]. Furthermore most of the topologies are not planar, and the classification becomes meaningless if (almost) all topologies are in the same class.

I want to suggest systematics based on the construction method of the topology. First it is possible to use well known simple structures as interconnection networks such as ring, line or tree. This method will be referred to as *simple connection structures*. The next family of topologies are *graphs on alphabets*, where the node addresses are words of specific length in a specific alphabet; a subgroup of this family are *hypercube structures*. Also some methods from group theory [Cayl 78] can be used in constructing static networks, the resulting graphs are called *Cayley Graphs*. In any graph which is not complete it is possible to decrease message delays and to improve fault tolerance by establishing *additional links*. Since the adding of links will increase connection costs it is advantageous to start with graphs which possess a low connectedness (trees, rings). An extension of the additional link concept are *Generalized Chordal Rings*, where links are added according to certain symmetry rules to a initially disconnected set of nodes. By the method *combination of basic modules* copies of (simple) structures are linked or merged[1] into more complex ones. The use of either identical or nonidentical subparts is possible. Another possibility for constructing more complex graphs out of some given ones are *boolean operations* on graphs. In the last section some *random methods* for constructing graphs are investigated.

When discussing a topology we will first give a (more or less) formal definition of the structure and then try to figure out the properties of the topology, mainly concentrating on

- connection costs, expressed in the total number of links,
- communication delays, in terms of diameter and average distance,

[1] To link structures means to establish links according to certain rules between the nodes of the subgraphs whereas merging stands for putting together subgraphs so that they share a set of nodes.

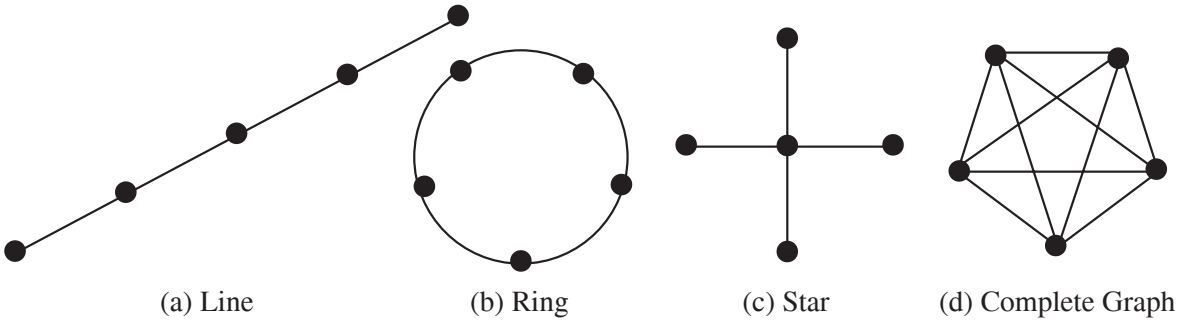


Figure 3.1: Simple connection structures

- fault tolerance, characterized by connectivity and cohesion,
- regularity and symmetry,
- ease of routing and
- extensibility.

If measures of the characteristics have a closed form (depending on the parameters of the topology), then those closed forms will be shown in a table at the end of every section comparing the topologies discussed in the section. [2]

3.1 Simple Connection Structures

Line L(N) The simplest way to connect N nodes is to arrange them in a line and connect neighbours. This results in lowest possible **connection costs** but also in highest possible **communication delays**. The **fault tolerance** is also poor, since the failure of a single node or link disconnects the complete structure. **Routing** and **extensibility** are trivial.

The practical use for multicomputer systems is limited to special purpose architectures (i. e. Pipelining). But the low fault tolerance poses a severe problem.

Ring C(N) [Agra 86] If the first and the last node in a line are connected (i. e. building a ring), we can achieve a reduction in **communication delays** (diameter 50% and average distance 75%), but the growth of diameter and average distance is still linear with the total number of nodes and becomes unacceptable for large values of N. However, if the degree is restricted to 2, lines and rings are the only possible topologies.

Routing is still simple and new nodes can be added anywhere in the ring, so we have good **extensibility**. The existence of 2 disjoint paths (one clockwise and on counter clockwise) between any two nodes increases the **fault tolerance**, rings are super-λ.

A further improvement (reduction in diameter and higher fault tolerance) can be achieved by adding links between distant nodes in a ring. A family of topologies using this construction technique are *Chordal Rings*, which will be discussed in section 3.5.

[2] A comparison and summary of all topologies is given in Section 4.

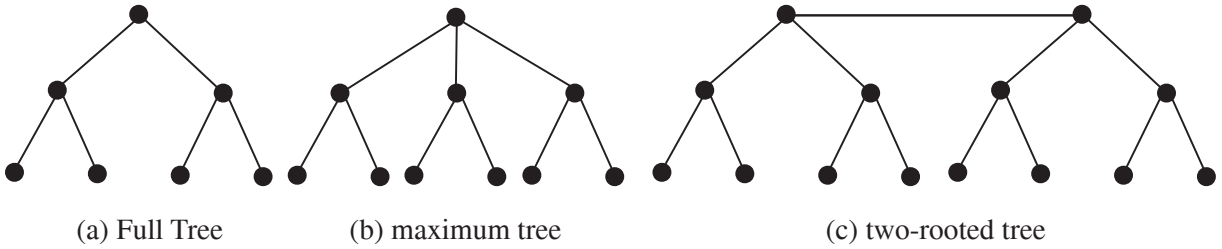


Figure 3.2: Tree topologies: $T^{full}(2,2)$, $T^{max}(2,2)$, $T^{2}(2,2)$

Completely Connected network K(N) If we establish a direct link between all pairs of processing elements, we obtain lowest possible **communication delays** and optimum **fault tolerance**, but the **connection costs** ($L = N(N-1)/2$) limit the practical use of completely connected networks.

Tree T(B, h) A lot of different types of trees have been defined, causing some confusion about the terminology, so we want to give the following definitions:

Definition 19 *A tree with B branches and height h, denoted by $T(B,h)$, is a graph with a special node called root, containing no cycle*[3]*, every nonleaf node has at most B children and the leaves are at most at distance h from the root node.*

A tree $T(B,h)$ is called balanced, denoted by $T^{bal}(B,h)$, if all leaves are exactly at distance h from the root node.

A tree $T(B,h)$ is called complete (or full), denoted by $T^{full}(B,h)$, if it is balanced and if every nonleaf node has exactly B children.[4]

A tree $T(B,h)$ is called maximum, denoted by $T^{max}(B,h)$, if it is balanced and every nonleaf node has a nodes degree of $B+1$.

A tree is called m-rooted, denoted by $T^{m}(B,h)$, if it consists of m subtrees and the root nodes of all subtrees are connected in a complete graph of m nodes.

The diameter in trees is determined by the sum of the distances from the two most distant leaves to the root, so in balanced trees k is given by $2h$. So it seems advantageous to use balanced trees because they will have smaller communication delays than an unbalanced tree of the same order.

Trees are hierarchical structures so they can easily be extended by adding new nodes next to the leaves (**extensibility** type 1).[5] The other possibility for increasing the number of nodes consists in adding more branches which has the advantage that the communication delays will remain small but for the costs of an increase in degree (extensibility type 5). In complete or maximum trees it is only possible to add a complete new level or branch (extensibility type 2 or 6).

The total number of links in a tree is given by $N-1$, so we have low **connection costs**, but we have to notice that trees are irregular structures and the node degree for nonleaf nodes grows with B.

[3] A cycle in a graph is a path (an alternating sequence of nodes and edges) starting at node x and ending at node x, where no edge occurs twice.

[4] That means that all levels are full.

[5] Note that the tree will become unbalanced.

	N	d	L	k	symmetry
$L(N)$	N	2	$N-1$	$N-1$	$\lceil \frac{N}{2} \rceil$
$C(N)$	N	2	N	$\lfloor \frac{N}{2} \rfloor$	1
$T^{full}(B,h)$	$\frac{B^{h+1}-1}{B-1}$	$B+1$	$N-1$	$2h$	$h+1$
$T^{max}(B,h)$	$\frac{B^h(B+1)-2}{B-1}$	$B+1$	$N-1$	$2h$	$h+1$
$S(N)$	N	$N-1$	$N-1$	2	2
$K(N)$	N	$N-1$	$\frac{N(N-1)}{2}$	1	1

	μ	μ_{norm}	extensibility type	routing type	λ
$L(N)$	$\frac{N+1}{3}$	$\frac{2N+2}{3}$	1	1	1
$C(N)$ N odd	$\frac{N+1}{4}$	$\frac{N+1}{2}$	1	1	2
$C(N)$ N even	$\frac{N^2}{4N-4}$	$\frac{N^2}{2N-2}$			
$T^{full}(B,h)$	$\frac{2B^{h-1}(h-1-\frac{2}{B-1}+\frac{2h}{B^h-1})}{B^{h-1}-1}$	$(B+1)\mu$	2 or 6	1	1
$S(N)$	$\frac{2(N-1)}{N}$	$\frac{2(N-1)^2}{N}$	5	1	1
$K(N)$	1	$(N-1)$	5	1	$N-1$

Table 3.1: Simple connection structures

The major disadvantages of trees are their poor **fault tolerance** (since there exists exactly one path between any two nodes, the topology can easily be disrupted), and the high message traffic density at the nodes next to the root and at the corresponding links [6], and decreases towards the leaves. In order to come up with these two problems several topologies have been proposed where additional links are established, for example between the nodes within a level (see Section 3.5).

For **routing** a message in a tree it is only neccessary to now the destination address of the message. A node receiving a message has to compare its address to the destination address. If they match the destination is reached, otherwise the node has to determine whether the destination node is in one of its subtrees or not. In the first case the message is routed downwards to the subtree containing the destination node, otherwise upwards. An appropriate labelling scheme for nodes enables fast comparison and subtree determination. In [Horo 81] a labelling scheme and a corresponding routing algorithm for binary trees is given.

The X-tree machine [Desp 78], the DAC machine [Horo 79] or the P-tree machine [Harr 79] are examples for using binary trees in the design of multiprocessor systems.

Star S(N) A star can be seen as an extremal tree with $h = 1$ and $B = N - 1$. The diameter of the structure is independent of the size of the network ($k = 2$), but the high node degree of the root node ($N - 1$) and the high number of messages passing through it poses a severe limitation on the size of this network.

Table 3.1 compares those simple structures. In trees it is possible to draw a tradeoff between degree and diameter, whereas in all other structures one parameter is independent of N and the other grows linear with N. When comparing communication delays we should use the normalized average distance since the topologies have different degrees.

Only rings and completely connected networks are regular 1-symmetric, and optimum fault tolerant (super-λ). We can see that all topologies offer an optimum routing algorithm.

[6] $\gamma^{node} = \frac{2(B^{h-1}-1)^2}{B-1}$ and $\gamma^{link} = \frac{2(B^{2h-2}-B^{h-1})}{B-1}$

3.2 Graphs on Alphabets

In this construction method the nodes are labelled by words of length n over an alphabet of b letters. A connection rule defines when to establish a link between two nodes. For all graphs on alphabets a similar routing algorithm can be derived based on a comparison of node addresses (computational routing). The main idea is to view routing as a sequence of communication steps equivalent to a sequence of changes made to the source address label to become the destination address label. A change in the address corresponds to the selection of a link for transmission. Since there is in general more than one possible path betweeen any two nodes, it is necessary to apply specific knowledge of the individual properties of the topology, in order to guarantee that the shortest path between any two nodes is found. We will not give explicit routing algorithms for the topologies in this section, but show routing in Boolean n-cubes by example in the next section. The Boolean n-cube belongs to the family of hypercube structures, which are graphs on alphabets, too, but will be discussed in a seperate section (3.3) because of their special properties.

Odd Graph OG(d) Akers [Aker 65] has proposed Odd Graphs as a network with a large number of nodes for small degree and diameter (see Section 5.1).

Definition 20 *An Odd Graph, denoted by $OG(d)$, consists of $N = \begin{pmatrix} 2d-1 \\ d \end{pmatrix}$ nodes, d denotes the number of 1's in the $(2d-1)$-bit binary node address ($d > 2$), where node u is connected to node v if and only if*

- *they have exactly one 1 in a common position.*

When constructing an Odd Graph, we first must choose the degree d. Then all nodes are labelled with those binary numbers of length $2d-1$, which contain exactly d 1's (For $d = 4$ there would be 35 nodes, as shown in figure 3.3 (a)). An edge is then drawn between each pair of nodes, whose assigned numbers have precisely one 1 in a common position. So it is guaranteed that each node has exactly d neighbours. This construction results in a regular and symmetric graph.

The diameter in an OG is given by $d-1$ and a computational **routing** algorithm finding this shortest path is given in [Aker 65].

The only possibility to increase the number of nodes in an Odd Graph consists in increasing the degree and since OGs are not hierarchical, they are of **extensibility type** 8. Since the degree grows with N, the **connection costs** must grow faster than linear with N.

de Bruijn Graph BG(n,b) [7] de Bruijn Graphs have been rediscovered since 1894 [Sain 94] several times using different approaches (see [Rals 82] for a detailed presentation). We want to follow the graph theoretic model proposed by de Bruijn [Brui 46], and further investigated by Pradhan and Reddy [Prad 82].

Definition 21 *A de Bruijn Graph, denoted by $BG(n,b)$, consists of b^n nodes, where n denotes the length of the node address in radix b representation ($n > 2$, $b > 2$), where node u is connected to node v if either*

- *the first $n-1$ digits of u are the last $n-1$ digits of v or*
- *the first $n-1$ digits of v are the last $n-1$ digits of u.*

[7] De Bruijn Graphs are also known under the name *shift connected vectors* [Cont 83] or *shuffle networks* [Prio 90].

The construction of de Bruijn Graphs requires the following steps: first we must choose upon b, the size of the alphabet, with the following limitation: b must not be less than 3 (otherwise the BG would be disconnected). We must further consider that the degree is determined by $d = 2b$, so if we have a restriction on d, b must not exceed $d/2$. The second parameter n (denoting the length of the words or node addresses) determines diameter $k = n$. Two nodes are connected, if the node address of one can be transformed into the node address of the other by a shift of one position, either to the left or to the right. The "empty" (rightmost or leftmost) position may be filled with any digit of the alphabet.

Due to this, the node degree is not constant for every node in the graph. In fact there are $N-b^2$ nodes with node degree $2b$, b nodes with node degree $2b-2$ and b^2-b nodes with node degree $2b-1$ and the structure is 3 symmetric since all nodes with the same node degree also have the same connection pattern. The **total number of links** is given by $L = Nb - (b^2+b)/2$. The **diameter** of the graph equals to n.

The line connectivity is determined by the lowest node degree in the de Bruijn Graph therefore $\lambda = 2b-2$. **Extension** of the graph can either be done by increasing the word length or the size of the alphabet. Clearly the first method increases diameter whereas the second one increases degree. Both methods require a restructuring of the existing network. A type 1 **routing algorithm** is given in [Prad 82].

Figure 3.3 (b) shows a $BG(4,2)$, and although BGs are 3-symmetric it was pretty difficult to arrange the nodes so that at least a little bit of symmetry can be seen.

Kautz Graph KG(n, b) [Kaut 68] A Kautz Graph $KG(n,b)$ is derived from a de Bruijn Graph with the same parameters $BG(n,b)$ by deleting letters with identical elements as neighbours. So the number of nodes in a $KG(n,b)$ is smaller the N in a $BG(n,b)$, both are of diameter n, but the KG has a smaller degree $d = 2(b-1)$.

Kautz Graphs have the same possibilities of **extension** as BGs and I suppose that it should be possible to derive an analogous **routing algorithm**, but I have not found one in the literature.

In figure 3.3 a $KG(3,3)$ is shown.

Moebius Graph MG(n) [Lela 82]

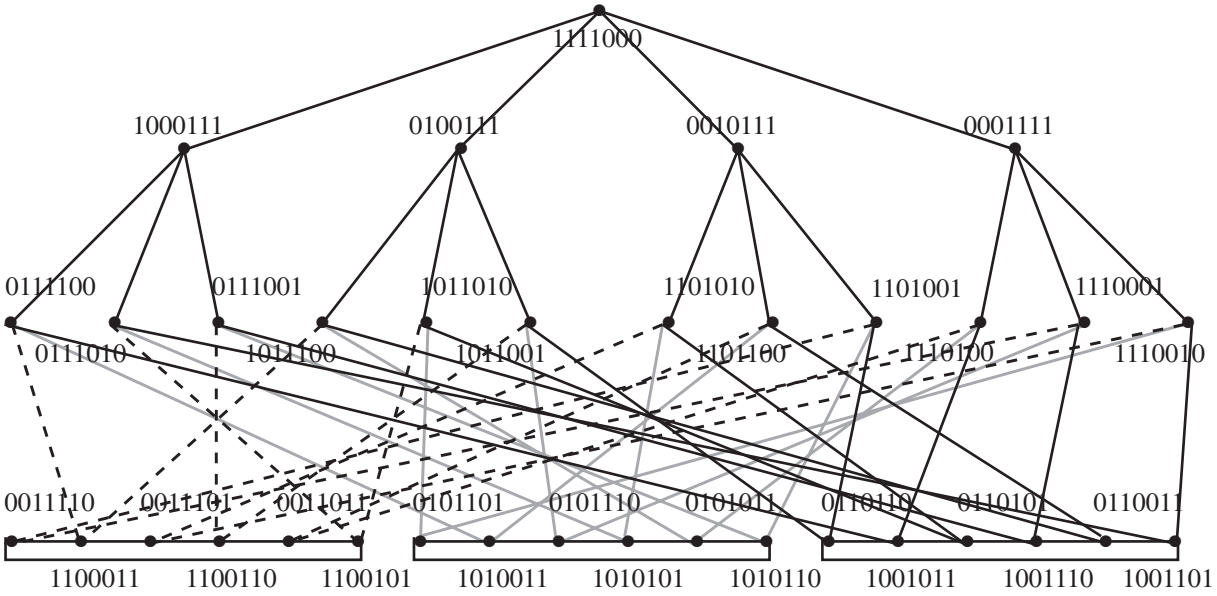
Definition 22 *A Moebius Graph, denoted by $MG(n)$, consists of 2^n nodes, where n denotes the length of the binary node address, where node u is connected to node v if*

- $u_0u_1\ldots u_{n-1} = v_1\ldots v_{n-1}\overline{v}_0$ *or*
- $u_0u_1\ldots u_{n-1} = \overline{v}_n v_1\ldots v_{n-1}$ *or*
- $u_0u_1\ldots u_{n-1} = v_0\ldots v_{n-3}\overline{v}_{n-2}\overline{v}_{n-1}$. [8]

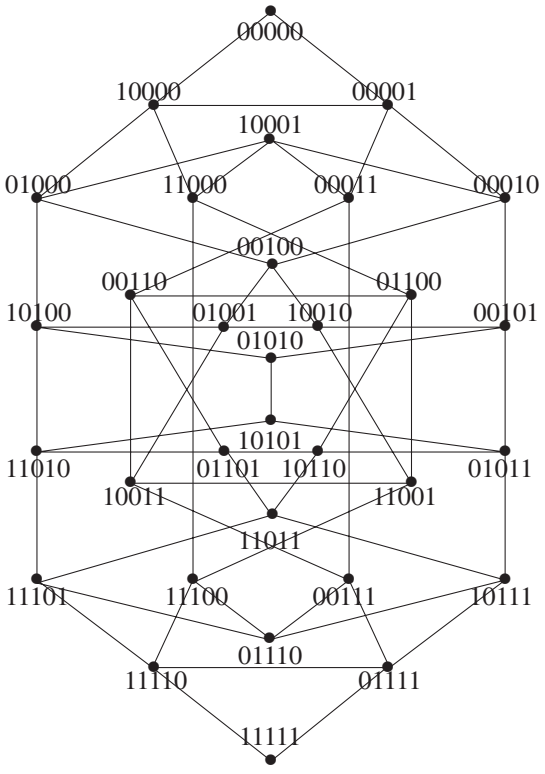
In contrast to the other graphs on alphabet described so far, Moebius Graphs have a constant **degree** of 3. The nodes are labelled with binary addresses and arranged in a ring. Two nodes are connected, if their node addresses can be transformed into each other by a cyclic shift to the left (right) and changing the rightmost (leftmost) digit into its complement (first 2 rules), or if their addresses differ exactly in the last two digits (third rule).

According to the first two rules, two twisted loops are formed (that is why the structure is called Moebius Graph), shown by the black and by the grey line in figure 3.3 (d)). According to the third rule an edge is drawn between those pairs of nodes whose addresses differ exactly in the rightmost two digits (black dashed lines).

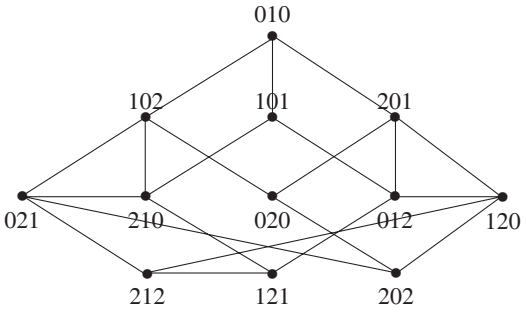
According to Leland [Lela 82] no exact closed form for diameter can be given, it is only known that $k < \lfloor 3n/2 \rfloor$. Although the diameter is small, the lack of a routing algorithm finding the shortest path between any two nodes deteriorates **communication delays**.



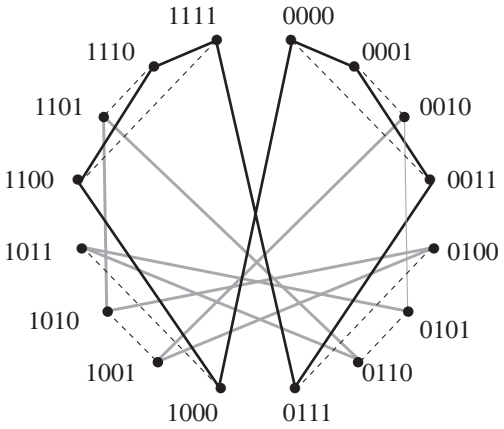
(a) Odd Graph



(b) de Bruijn Graph



(c) Kautz Graph



(d) Moebius Graph

Figure 3.3: Graphs on alphabets: $OG(4)$, $BG(5,2)$, $KG(3,3)$, $MG(4)$

	N	d	L	k	symmetry
Odd Graph	$\binom{2d-1}{d}$	d	$\frac{dN}{2}$	$d-1$	1
deBruijn	b^n	$2b-2 \leq d_i \leq 2b$	$Nb - \frac{b^2+b}{2}$	n	3
Kautz	$b(b-1)^{n-1}$	$d_i \leq 2(b-1)$		n	
Moebius	2^n	3	$\frac{3 \cdot 2^n}{2}$	$\leq \lfloor \frac{3n}{2} \rfloor$	1

Table 3.2: Graphs on alphabets

There are a lot of other graphs on alphabeths. We have chosen these 4 graphs as representatives, because they show the variety of possible constructions for graphs on alphabets.

Moebius and Odd Graphs are regular, whereas the two others are irregular. Moebius Graphs are cubic graphs (graphs of degree 3). The Kautz and de Bruijn Graphs offer the possibility to choose two parameters, one determing the degree, and the other determing the diameter. Odd Graphs posses only one parameter, determining both, degree and diameter.

The connection costs are lowest in Moebius Graphs and maximum in Odd Graphs, the advantage of those graphs is their regularity and symmetry[9] (see table 3.2).

3.3 Hypercube Structures

This section deals with construction methods based on an n-dimensional hypercube. Every node is identified via its coordinates in the positive quadrant of an n-dimensional Euclidean space. Note that these topologies are graphs on alphabets, but I have chosen to devote them a seperated section because of their importance.

3.3.1 Binary Hypercubes

All topologies in this section consist of $N = 2^n$ nodes, arranged in a hypercube of dimension n and differ from each other in having different connection rules. These different connections result in different values for diameter and average distance.

All binary hypercubes are regular of degree n and have therefore a total number of $L = \frac{N \log_2 N}{2}$ links, so the **connection costs** grow more than linear with N. All binary hypercubes are optimum **fault tolerant** (super-λ).

Boolean n-cube Q(n) [10] [Witt 76],[Peas 77] The most popular structure for multicomputer interconnection is probably the Boolean n-cube, favoured because of its regularity and symmetry and high potential for exploiting parallelism.

A rich literature deals with the anlysis of Boolean n-cubes, ([Sull 77], [Witt 81], [Efe 91]) fault diagnosis ([Arms 81], [Yang 88]) and routing ([Ho 86], [Chen 90]). Some existing parallel machines use this structure as interconnection network (Cosmic cube [Seit 85],...).

Definition 23 *A Boolean n-cube, denoted by $Q(n)$, consists of 2^n nodes, where u is a n-digit binary node address, where node u is connected to node v if and only if*

- *their node addresses differ in exactly one digit.* [11]

[8] u_i denotes the i-th digit of the binary node address and $\overline{u}_i$ denotes the binary complement.

[9] Unfortunately, I was not able to draw the graph in a way which shows the symmetry.

[10] This structure is often discussed under the name "hypercube", which we use as generic term here.

[11] A more elgant recursive definition is given in Section 3.8.

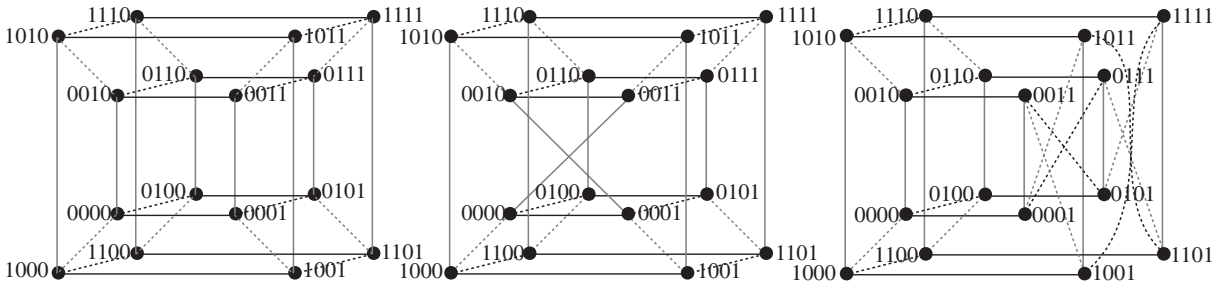


Figure 3.4: Binary hypercubes: $Q(4)$, $TQ(4)$, $MQ(4)$

Diameter and degree in a $Q(n)$ are given by n, so when increasing the size of a $Q(n)$ both parameters grow at the same rate. The Boolean n-cube is often favoured for its low communication delays, but if we also take the high connection costs into account, it becomes less attractive. Nevertheless, it is frequently used as interconnection topology, because a lot of parallel algorithms can be (and have been) easily implemented on hypercubes.

Routing in Boolean n-cubes requires only the knowledge of the source and destination addresses. Those addresses are compared, and if they match, the destination is reached. Otherwise the message is sent into the first dimension, where a mismatch occured. The number of mismatches (differing digits) determines the length of the path. The following example shows the routing sequence (list of node addresses), when routing a message in a $Q(7)$ from node 0011001 to node 1111000:

	0	0	1	1	0	0	1	mismatch at the 1^{st} position
$\rightarrow$	1	0	1	1	0	0	1	mismatch at the 2^{nd} position
$\rightarrow$	1	1	1	1	0	0	1	mismatch at the last position
$\rightarrow$	1	1	1	1	0	0	0	destination reached

An improvement in diameter of n-cubes can be achieved by breaking their symmetry (i. e. not every node is connected to its direct neighbours). In the following two structures in this section links are rearranged according to certain rules (*twist operation*) in order to reduce diameter.

Twisted Cube TQ(n) [Esfa 88] Esfahanian discovered the ability of reducing the diameter in a Boolean n-cube by performing only one single twist operation between any two node disjoint edges on a shortest cycle in a hypercube.

Definition 24 *An n-dimensional Twisted Cube, denoted by TQ_n, is made of a Boolean n-cube by replacing the edges (x_1, x_2) and (y_1, y_2), which must be two disjoint edges on a shortest cycle*[12] *in a $Q(n)$, by the edges (x_1, y_2) and (y_1, x_2).*

It is sufficient to exchange the two edges of any lateral surface, as shown in figure 3.4, in order to reduce diameter by 1 ($k = n - 1$). The disadvantage of the reduction in diameter is a loss in symmetry which complicates **routing**.

Multiply-twisted Cube MQ(n) [Efe 88],[Hilb 87]

[12] Any four edges in a $Q(n)$, which form a square, are a shortest cycle.

	N	d	L	k	symmetry	extensibility Type	routing Type
$Q(n)$	2^n	n	$\frac{N \log_2 N}{2}$	n	1	7	1
$TQ(n)$	2^n	n	$\frac{N \log_2 N}{2}$	$n-1$	$\frac{N}{2}$	7	2
$MQ(n)$	2^n	n	$\frac{N \log_2 N}{2}$	$\lceil \frac{n+1}{2} \rceil$	$n-1$	7	2

Table 3.3: Binary hypercubes

Definition 25 *A one-dimensional multiply-twisted Cube, denoted by MQ_1, consists of two nodes 0 and 1 with one edge between them.*

In a MQ_n two multiply-twisted Cubes of dimension $n-1$ (MQ^0_{n-1} and MQ^1_{n-1}) are linked according to the following rule: $0u_{n-2}\cdots u_0 \in MQ^0_{n-1}$ is adjacent to $1v_{n-2}\cdots v_0 \in MQ^1_{n-1}$ if and only if

- *$u_{n-2} = v_{n-2}$ if n is even and*
- *for $0 \leq i \leq \lfloor \frac{n-1}{2} \rfloor$: $u_{2i+1}u_{2i} \sim v_{2i+1}v_{2i}$*

$(x \sim y)$ denotes two pair related binary strings, where $(x \sim y)$ if and only if $(x,y) \in \{(00,00),(10,01),(01,11),(11,01)\}$.

An $MQ(n)$ can be recursively constructed of two $MQ(n-1)$, similar to Boolean n-cubes, but on one side the edges are twisted as shown in figure 3.4. An $MQ(n)$ achieves even more reduction in **diameter** than the $TQ(n)$, $k = \lceil \frac{n+1}{2} \rceil$.

A **routing algorithm** finding the shortest path between any two nodes is given in [Efe 88].

The major drawback of all binary hypercubes is, that the degree increases when the network size is increased, which results in poor extensibility, although the topologies are hierarchical (see table 3.3). The MQ has the smallest communication delays, but routing is a little bit more complicated.

3.3.2 Generalized Binary Hypercubes

Binary hypercubes are often favoured for their low values in diameter and average distance. But the major drawback is an increase in degree of order $\log_2 N$. Values for normalized average distance are substantially worse compared to other topologies. And if the degree is limited, also the number of nodes in the binary hypercubes is bounded. The following two structures try to cope with this problem by replacing a single node in a binary hypercube by a ring of nodes, so the total number of links is given in both structures by $N = R2^n$ where R denotes the number of nodes within a ring.

Cube-Connected-Cycles CCC(n) [Prep 81],[Carl 85]

Definition 26 *A Cube-Connected-Cycles network of dimension 1, denoted by $CCC(1)$, consists of 2 nodes ("cycle" of length 1) with one link between them.*
A $CCC(n)$ is made of 2 $CCC(n-1)$ by inserting exactly one node in every ring at the same position and linking the two CCC via a link between each pair of the corresponding new nodes.

Figure 3.5 shows the recursive construction of a CCC. The nodes are addressed by a pair $(r|i)$ where r is called the ring address and i is the node address. All links connecting nodes with addresses $(.|i)$ are called *sheaf i*. A CCC is regular with constant degree 3 and node symmetric. **Diameter** and other measures are analysed in [Raab 88] and [Agra 86], the results are shown in table 3.4.

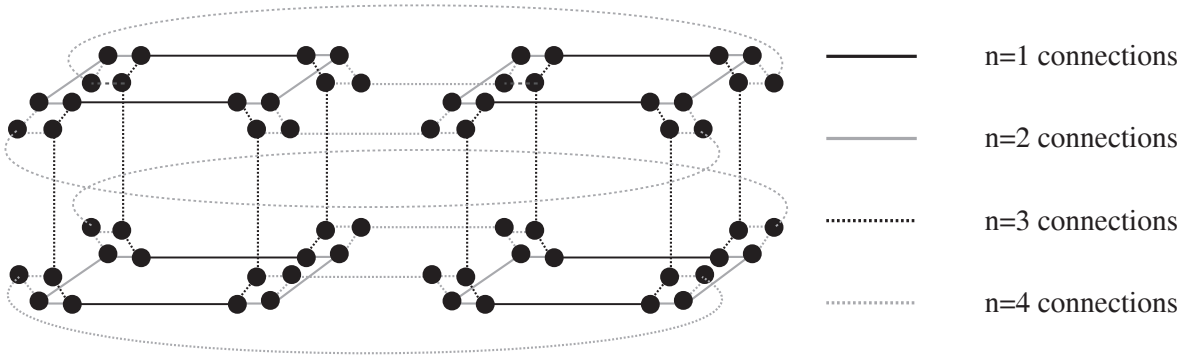


Figure 3.5: Construction of a $CCC(4)$

When **routing** a message in a CCC, the source and destination addresses are compared. The number of differing digits in the ring addresses indicate, how often the rings must be changed (at most n times, the length of the ring address). The position of the differing digit indicates, which sheaf link must be used in order to change to the next ring. When ever the destination node is not in this sheaf, it is necessary to change sheafs, too. [13]

A generalization has been made in order to allow other values for N by adding nodes to each ring without increasing the dimension of the CCC:

Definition 27 *A Generalized Cube Connected Cycles network of dimension n, denoted by (GCC_n), has $N = R2^n$ nodes, R denotes the number of nodes in every ring, $R \geq n$, $(r|i)$ is the node address with r as an n-digit binary ring address and i denotes the number of the node in the ring. Node $(r|i)$ is connected to node $(s|j)$ if and only if*

- *$r = s$ and $i \oplus_R 1 \equiv j$ (connection within a ring) or*
- *$i \leq n$ and $i = j$ and the ring addresses differ only in the i-th bit (connection between the rings).*

A GCC is irregular with $d_1 = 3$ for all nodes with addresses $(r|i \leq n)$ and $d_2 = 2$ for all other nodes $(r|i > n)$.

A comparison between CCC's and Shuffle-Exchange graphs can be found in [Leng 82]; Jain [Jain 88] has shown an isomorphism between CCC and HCSN[14].

Generalized Boolean n-cube GQ(R, n) [Huan 88] In Generalized Boolean n-cubes it is possible to draw tradeoff between degree and diameter, because the topology is specified by two parameters, one determing the degree of the structure, the other determing its diameter.

Definition 28 *A Generalized Boolean n-cube, denoted by $GQ(R, n)$, consists of $N = R2^{n-1}$ nodes, where $(r|i)$ denotes the node address with r as an $n-1$ digit binary ring address and $i = 0, 1, \ldots, R-1$ denoting the number of the node in the ring, where node $(r|i)$ is connected to $(s|j)$ if and only if*

- *$r = s$ and $i \oplus_R 1 \equiv j$ or*
- *$i = j$ and the ring addresses differ in exactly one bit.*

[13] A detailed description of this routing algorithm can be found in [Raab 88].

[14] A HCSN is a Homogeneous Circular Shuffle Network proposed by [Trip 85].

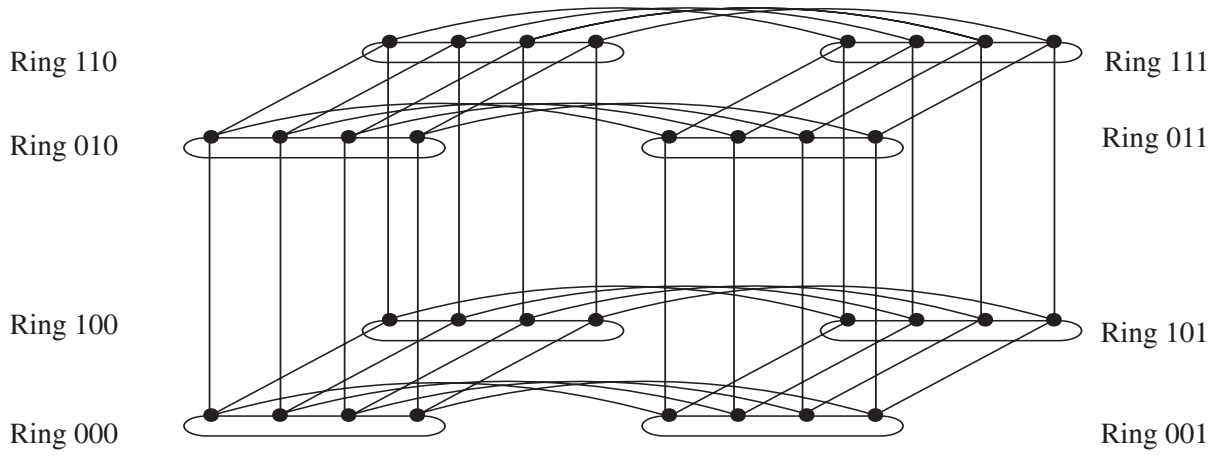


Figure 3.6: Generalized Boolean n-cube

As we can see this structure is very similar to CCC's. It is regular of degree $n+1$ (2 links to the neighbours in the ring and $n-1$ links to nodes in other rings). A GQ is completely node symmetric and offers a maximum connectivity (see [Huan 89] for a detailed analysis of **reliability**).

3.3.3 Variation of W

We now want to investigate hypercube-like topologies with radix $W > 2$. The extension to a higher radix allows an increase in the number of nodes without an increase in the dimension (node degree). There are two different rules for connecting nodes:

1. A node is connected to those nodes whose n-digit node addresses differ in only one digit by 1 (Meshes, Tori).
2. A node is connected to those $n \cdot (W_i)$ nodes whose n-digit node addresses differ exactly in the i-th digit (W-ary n-cube, Generalized Hypercubes).

Mesh M(W, n) [Bhat 82]

Definition 29 *An n-dimensional mesh, denoted by $M(W,n)$, consists of W^n nodes, where $\vec{u}$ denotes the coordinates of a point in an n-dimensional coordinate system with $u_i \in 0, 1, \ldots, W-1$, where node $\vec{u}$ is connected to node $\vec{v}$ if and only if*

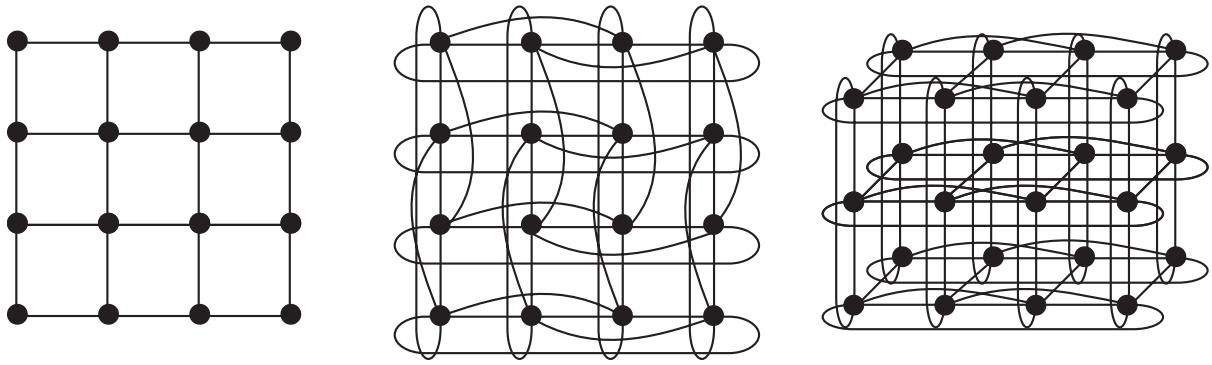
- *$v_i = u_i + 1$ in dimension i and $v_j \neq v_j$ in all other dimensions.*

If the addition is taken modulo W then also node 0 is connected to node $W-1$, resulting in a mesh with wrap around links, also called *Torus* [Seit 84], [Dall 86].

Meshes are irregular, since the nodes at the border have a smaller node degree than inner nodes. Tori are regular and have furthemore smaller **communication delays**. Routing is in both structures similar to routing in Boolean n-cubes.

W-ary n-cube HC(W, n) [Hawk 85], [Dall 90]

Definition 30 *A W-ary n-cube, denoted by $HC(W,n)$, consists of W^n nodes, where u is an n-digit radix W node address, where node u is connected to node v if and only if*



(a) Mesh (b) W-ary n-cube (c) Generalized Hypercube

Figure 3.7: Hypercube-like structures: $M(4,2)$, $HC(4,2)$, $GHC((4,2,3))$

- $v_i \equiv u_i \oplus_W w$ *in dimension* i *for any* $w \in \{1,2,\ldots,r-1\}$ *and* $v_j \neq v_j$ *in all other dimensions*

This definition is a generalization of the Boolean n-cube. Insterad of using binary node addresses, radix W labels are used. The **connection costs** are higher $d = (W-1)n$, the **diameter** is determined by $k = n$ just as in Boolean n-cubes. The topology is regular and node symmetric (see also table 3.4).

Generalized Hypercube GHC($\vec{\mathbf{m}}$) [Bhuy 82], [Bhuy 84] In a GHC the number of nodes on each coordinate axes are allowed to vary, so the name "*hyperquader*" would be more appropriate. [15]

Definition 31 *A Generalized Hypercube, denoted by* $GHC(\vec{m})$*, consists of* $N = \prod_{i=1}^{n} m_i$ *nodes, where* m_i *denotes the number of nodes in the* i*-th dimension and* u *is a* n*-digit node address with* u_i *denoting the* i*-th digit of* u *with* $0 \leq u_i \leq (m_i - 1)$*, where node* u *is connected to node* v *iff*

- *they have exactly one differing digit.*

This structure is regular of degree $d = \sum_{i=1}^{n}(m_i - 1)$ and node symmetric, values for diameter and average distance are shown in table 3.4. The topology remains connected unless more then $d-1$ links fail and is therefore super-λ.

Since there are several different partitionings for a given number of nodes one might be interested in finding an optimum structure. One optimization is to minimize the total number of links for given diameter, which is achieved if all m_i are as close as possible to $\sqrt[n]{N}$ (see [Bhuy 84] for a proof). Another approach tries to minimize the product of diameter and degree in order to find a compromise between degree and diameter.

A point-to-point routing algorithm and a single-source broadcast algorithm are given in [Bhuy 84].

[15] This structure is also known under the name *Alpha network* [Agra 86].

	N	d	L	k	symmetry
$CCC(n)$	$n2^n$	3	$\frac{3N}{2}$	$\lfloor\frac{5n-2}{2}\rfloor$	1
$GCC(R,n)$	$R2^n$	3	$(2R+n)2^{n-1}$		R
$GQ(R,n)$	$R2^{n-1}$	$n+1$	$R(n+1)2^{n-2}$	$\lfloor\frac{R}{2}\rfloor+n-1$	1
$HC(W,n)$	W^n	$(W-1)n$	$\frac{(W-1)nW^n}{2}$	n	1
$GHC(\vec{m})$	$\prod_{i=1}^{n} m_i$	$\sum_{i=1}^{n}(m_i-1)$	$\frac{N\sum_{i=1}^{n}(m_i-1)}{2}$	n	$>n$
$M(W,n)$	W^n	$2n$	$<nW^n$	$n(W-1)$	$Wn/2$

	μ	λ	extensibility Type	routing Type
$CCC(n)$	$\frac{7n}{4}-3+\frac{n+1}{2^{n-1}}$	d	4	2
$GCC(R,n)$		$d-1$	3	2
$GQ(R,n)$	$\frac{2^{n-3}R(R+2n-2)}{(R2^{n-1}-1)}$ for even R $\frac{2^{n-3}(R^2+2(n-1)R-1}{R2^{n-1}-1}$ for odd R	d	4 or 8	1
$HC(W,n)$	$\frac{nW^{n+1}}{4(N-1)}$ for even W $\frac{nW^{n+1}(1-W^{-2})}{4(N-1)}$ for odd W	d	6	1
$GHC(\vec{m})$		d	2 or 6	2
$M(W,n)$		$d-n$	2 or 6	2

Table 3.4: Hypercubes

3.4 Cayley Graphs

In this section we examine a large and versatile family of graphs, called Cayley Graphs [Cayl 78] or group graphs [Aker 84]. Those graphs offer an (almost) unlimited variety for constructing interconnection topologies on the one hand, but can all be defined in precisely the same way by simply specifiying a set of basic transformations. We will first give a formal definition and discuss general properties, possessed by all Cayley Graphs.

Definition 32 *The nodes of a Cayley Graph*[16]*, denoted by CG, are the elements of a finite group $\mathcal{G}$, where node g is connected to node $g*\omega$, Ω denotes the set of generators ω for $\mathcal{G}$ with $\omega\in\Omega$ if and only if $\omega^{-1}\in\Omega$ and the identity element of $\mathcal{G}$ is not in Ω, $*$ is the group multiplication.*

A Cayley Graph results, when we select a set of n symbols and a set of rules (elements of Ω) by which one permutation of the symbols can be changed into another. The nodes are labelled with permutations on the n symbols and an edge is drawn between two nodes if there is a rule by which the label of one node can be changed into the label of the other. The rules (or transformations) are permutations themselves.

A simple example is the following: consider a set of symbols $\{A,B,C,D,E\}$, and a single transformation rule $\omega=(bcdea)$ (and $\omega^{-1}=aedcb$), a cyclic shift. [17] The resulting structure is a ring of 5 nodes. If the set of transformation rules is equal to G (i. e. all permutations are allowed), the resulting structure is a complete graph).

Arden found the following properties, which are inherent to all Cayley Graphs [Arde 89]:

- Cayley Graphs are undirected.[18]

[16] also called *Group graphs* in [Aker 84].

[17] We will denote symbols by capital letters, permutations by lower case letters.

[18] If ω^{-1} is not in Ω the resulting graph would be directed.

- Cayley Graphs are connected.
- Cayley Graphs are node symmetric.
- A node will not be connected to itself. [19]
- The number of nodes is bounded by the number of possible permutations $N \leq n!$.
- The degree is given by the number of generators, $d = \|\Omega\|$.
- **Routing** in Cayley Graphs consists in sorting a given permutation (the address of the source node) into another one (address of the destination node).

Most of the node symmetric structures can be described in terms of Cayley graphs. Among those structures are Boolean n-cubes, Generalized Boolean n-cubes (as shown in [Huan 89]) or Cube-Connected-Cycles (as shown in [Carl 85]). A famous node symmetric structure that cannot be represented as a Cayley graph is the Petersen graph [Hoff 60].

In the sequel, special permutations (swaps, flips and cyclic shifts) and the corresponding Cayley Graphs will be investigated. We will only show the permutations ω, an will keep in mind that also ω^{-1} is an element of Ω.

3.4.1 Swaps

A special type of a permutation is a swap, the exchange of two single symbols. The set of swaps can be represented as a graph on n nodes and a link between two nodes if and only if the exchange of these two symbols is allowed. [20] This graph is called a *transposition tree* [Aker 86].

Star Graph SG(n) [Aker 87], [Akl 90] In Star Graphs the swap of the first element to any of the $n-1$ other elements is allowed. It is called Star Graph since its transposition tree is a star.

We can use the following algorithm for **routing**, which tries to swap the symbol on the leftmost position in the source address to its position in the destination address. If the position in the destination address is also the leftmost then the algorithm is blocked and we have to make a wasteful step by exchanging this symbol to any other symbol which is not at its final place. Then the algorithm can continue. We will show as an example the route from node $FABCDGE$ to node $BCDAFEG$ in a $SG(7)$:

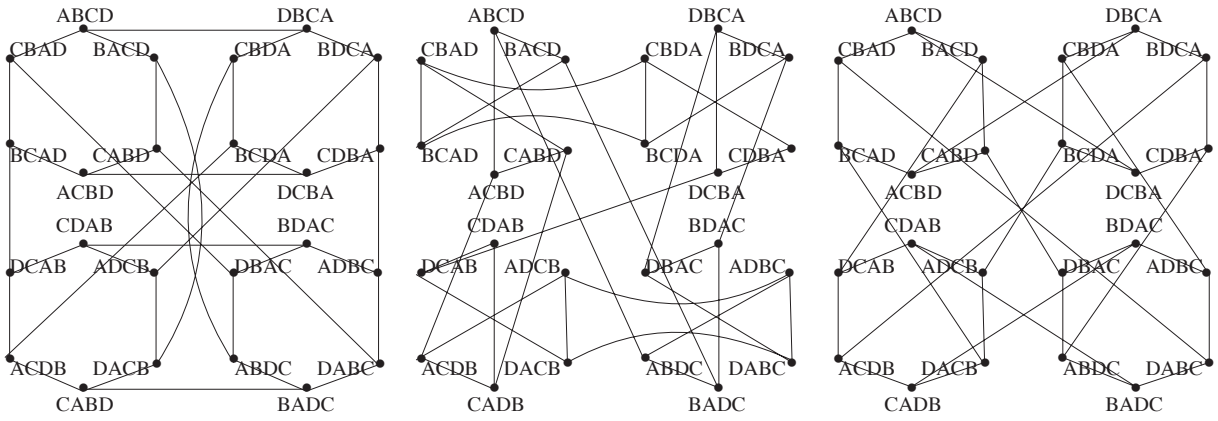
	F	A	B	C	D	G	E	swap F and D
$\rightarrow$	D	A	B	C	F	G	E	swap D and B
$\rightarrow$	B	A	D	C	F	G	E	swap B and A (redundant!)
$\rightarrow$	A	B	D	C	F	G	E	swap A and C
$\rightarrow$	C	B	D	A	F	G	E	swap C and B
$\rightarrow$	B	C	D	A	F	G	E	swap B and G (redundant!)
$\rightarrow$	G	C	D	A	F	B	E	swap G and E
$\rightarrow$	E	C	D	A	F	B	G	swap C and B
$\rightarrow$	B	C	D	A	F	E	G	destination reached!

Bubble Sort graph BSG(n) [Aker 84], [Aker 86] In a BG the exchange of any two *adjacent* symbols is allowed, so the transposition tree of a BSG is a line of length $n-1$.

A path from one node to any other node in a BSG is a sequence of adjacent transpositions and the **routing algorithm** for BG's is based on the well known bubble sort algorithm (modified slightly in order to sort a given permutation into another arbitrary permutation instead of an increasing sequence). Let us consider as an examplef a message from node $ABCD$ to node $BCDA$ in a $BSG(4)$:

[19] Since the identity element of $\mathcal{G}$ is not in Ω.

[20] Note that this graph is **not** the topology, but merely a representation of its construction.



(a) Star Graph (b) Bubble Sort Graph (c) Pancake Graph

Figure 3.8: Cayley Graphs: $SG(4)$, $BSG(4)$, $PG(4)$

	A	B	C	D	swap A and B
→	B	A	C	D	swap A and C
→	B	C	A	D	swap A and D
→	B	C	D	A	destination reached!

3.4.2 Flips

Pancake Graph PG(n) [Aker 84] Pancake Graphs arise from the *pancake flipping problem*, an old combinatorial problem which consists in ordering (by size) a stack of pancakes by flipping top substacks with a spatula. The number of pancakes corresponds to the number of symbols n and the generators are the f-flips[21].

Finding the **diameter** of a Pancake Graph is equivalent ot the problem of finding the minimum number of flips necessary to sort a stack of pancakes. This problem has been discussed for instance in [Gate 79], but has not been solved yet. In table 3.5 bounds for best values found so far are shown (taken from [Aker 86]).

A recursive **routing algorithm** can be derived easily: We start with two node addresses of length n and check whether the rightmost symbol in the destination address is at the rightmost place of the source address. If so, then this symbol is deleted from both addresses and the recursion continues with addresses of length $n-1$. If not, we will find the symbol at any other position in the source address, and at most two flips are necessary in order to bring it at the rightmost position, and we can continue the recursion as in the first case. It is easy to show that the number of routing steps will be at most $2(n-1)$. Take for an example the path from node $ABDEC$ to node $BCEDA$:

	A	B	D	E	C	5-flip
→	C	E	D	B	A	3-flip
→	D	E	C	B	A	4-flip
→	B	C	E	D	A	destination reached!

3.4.3 Cyclic Shifts

A cyclic shift moves all symbols about one position and the out slipping symbol on one end is inserted at the released position on the other end. Allowing only a left- or right cyclic shift will

[21] Performing a f-flip means to turn over a substack containig leftmost ("top") f elements.

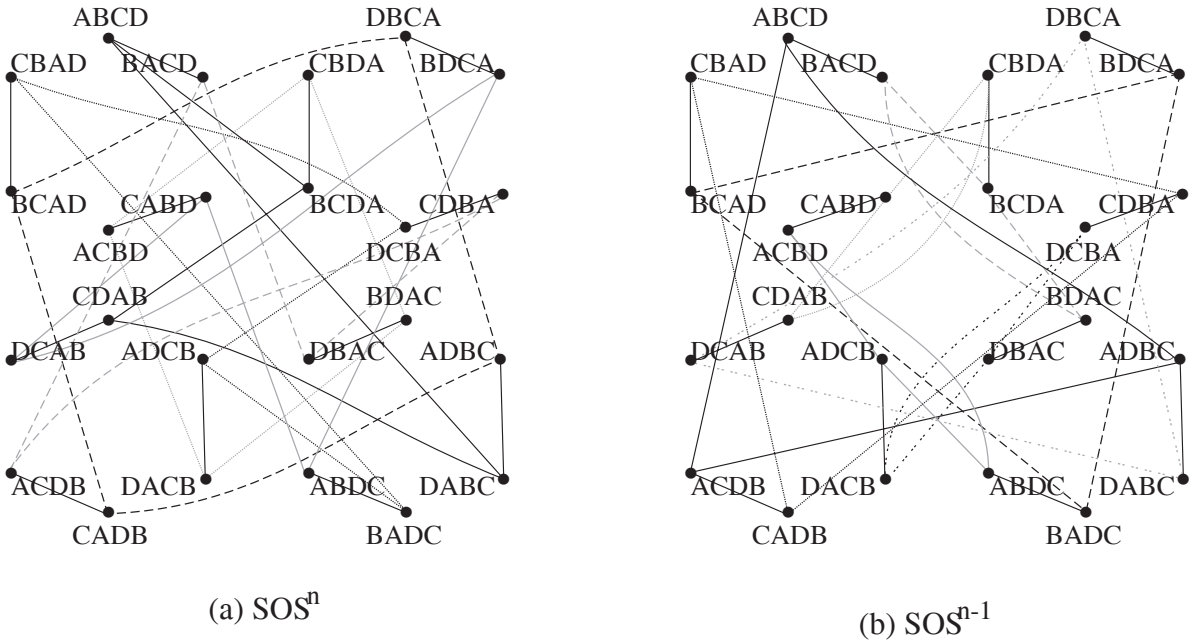


Figure 3.9: Cayley Graphs: $SOS^n(4)$, $SOS^{n-1}(4)$

result in a simple ring structure. The combination of shifts and swaps yields more interesting results:

SOS$^{\mathbf{n}}$(n) [Aker 84] In a $SOS^n(n)$ graph (swap or shift) it is allowed to swap the two left most symbols or to perform a cyclic shift of all n symbols either to the right or to the left. The resulting structure is shown in figure 3.9 (a).

SOS$^{\mathbf{n-1}}$(n) [Aker 84] If only the rightmost $n-1$ symbols are shifted (to the left or right) and the leftmost symbols may be swapped, we can achieve a reduction in diameter. The resulting structure is shown in figure 3.9 (b).

Cayley Graph have the advantage of beeing node symmetric. This means, that **routing** algorithms can be defined more easily, although these algorithms will not always find the shortest path between any two nodes. Another important feature is their optimum **fault tolerance**. But further research in this area is required, including the development of fault tolerant routing algorithms for Cayley Graphs in order to make use of their high fault tolerance.

The two SOS graphs are regular of degree 3 and have therefore lower **connection costs**, than the others. But **diameter** in Pancake and Star Graphs is smaller.

The **extensibility** is poor, for BG, PG and SG since the degree must be increased.

3.5 Additional Links

If we take an arbitrarily topology and add edges to it, it will usually be possible to reduce communication delays and to increase fault tolerance. The disadvantage is an increase in the connection costs. So it is advantageous to use topologies with small degree or irregular topologies. The adding of links to those nodes in irregular structure with smaller node degree, will remain the degree d unchanged unless the number of additional links to node i with node degree d_i will not exceed $d - d_i$.

In the following we will show several structures, trying to improve trees, rings and hypercubes.

	N	d	L	k	symmetry
BSG	$n!$	$n-1$	$(n-1)n!/2$	$\binom{n}{2}$	node
PG	$n!$	$n-1$	$(n-1)n!/2$	$17n/16 \leq k \leq 5(n+1)/3$	node
SOS^n	$n!$	3	$3n!/2$	$\binom{n}{2}$	node
SOS^{n-1}	$n!$	3	$3n!/2$	$\binom{n-1}{2} \leq k \leq \binom{n}{2}$	node
SG	$n!$	$n-1$	$(N-1)n!/2$	$\lceil 3(n-1)/2 \rceil$	node

	μ	κ	extensibility Type	routing Type
$BSG(n)$		$n-1$	8	4
$PG(n)$		$n-1$	8	4
$SOS^n(n)$		3	8	3
$SOS^{n-1}(n)$		3	8	3
$SG(n)$	$n+2/n+H_n-4$ H_n ...n-th Harmonic Number	$n-1$	6	2

Table 3.5: Cayley Graphs

Hypertree HT(h) Goodman [Good 81] proposed a topology based on binary trees with additional links among nodes within the same level.

Definition 33 *A Hypertree, denoted by $HT(h)$, is a binary tree $T^{full}(2,h)$, with an additional link between node u and node v[22] if and only if*

- *u and v are in the same level l and*
- *their node addresses differ exactly in digit $i = \frac{l}{2^{z+1}+\frac{1}{2}}$.*

z denotes the number of consecutive trailing zeros in l ($0 \leq l \leq h$) and i denotes the i-th bit in a node address (starting with the leftmost bit as zero).

The construction of a hypertree starts with choosing the height of the tree. After the tree connection have been established, additional links between nodes at the same level are established, starting at the first level. Since the goal is a reduction in diameter, we will connect those nodes, which are most distant so far. This is guaranteed by the second condition in the definition.

By definition the root node and all leaf nodes have a node degree of 2, all other nodes have $d_i = 4$, so the **connection costs** are higher than in a binary tree, which is the price for the reduction in **communication delays**. There is a reduction in diameter of about 25% in contrast to a binary tree and also the fault tolerance is improved, because even a faulty root node will not disconnect the structure.

A **routing algorithm** combining the ideas of routing in trees (messages are rooted upwards until the the destination is a direct descendent of the current position) and in hypercubes (comparison of node addresses) which will always be able to find the shortest path between any two nodes has been derived [Good 81].

[22] The usual labelling scheme for binary trees is used where all nodes have binary addresses, the root node has address 1, the left child of node u has label $u0$ and the right child has label $u1$.

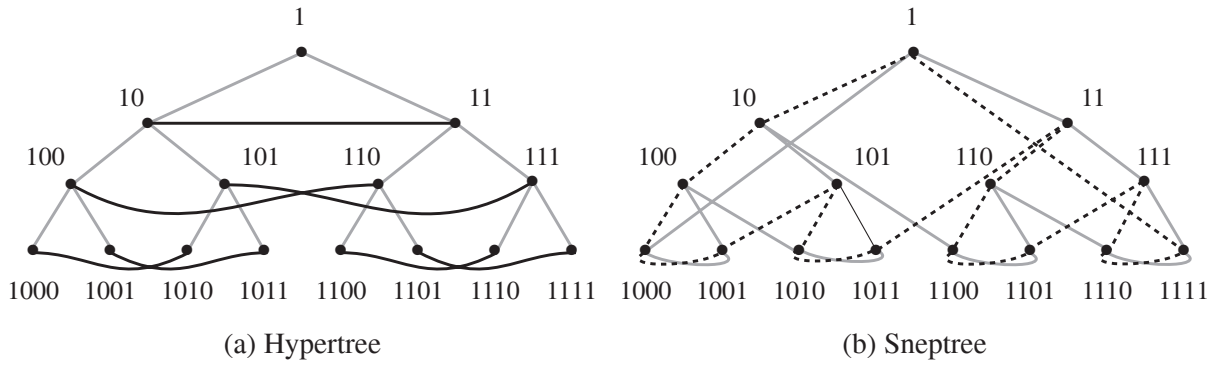


Figure 3.10: Trees with additional links: $HT(3)$, $ST(3)$

The **extension** of a hypertree is simply done by adding a new level (increasing h). The existing structure need not be disrupted, since the additional links are only established between nodes in the same level.

Sneptree ST(h) [Li 86], [Mart 90] In contrast to hypertrees, the additional links in a sneptree may be established to any other node in the tree, which fulfills the condition, that its node degree d_i is smaller then the degree of the topology.

Definition 34 *A Sneptree, denoted by $ST(h)$, consists of a binary tree $T^{full}(2,h)$ and $4-d_i$ additional links at node i where d_i denotes the node degree of node i in T^{full}.*

There is no explicit rule given which nodes have to be connected with the additional links and it can be shown that there are about $((2^{h-1}-1)!)^2$ different connection patterns [Li 86]. The only restriction is, that every node must become full, i. e. have node degree d.

If the additional links are chosen in a way, that there are two disjoint spanning cycles [23] in the tree, one containing only left links, the other containing only right links (see figure 3.10 (b)) [24], then the sneptree is called *cyclic*. A cyclic sneptree has the advantage of beeing symmetric and easier to extend. When adding a new level, only the links of the last level must be rearranged.

The **communication delays** are smaller than in binary trees, and also the **fault tolerance** is improved, the **connection costs** grows only linearly with N.

Chordal Ring CR(w) [Arde 81], [Skil 85]

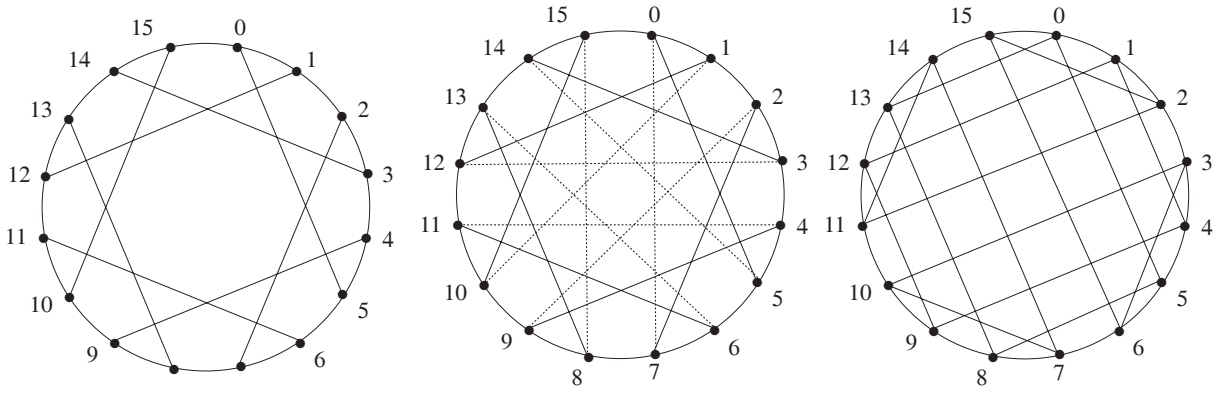
Definition 35 *A Chordal Ring, denoted by $CR(w)$, consists of a ring of N nodes $C(N)$ with nodes labelled $(0,1,\ldots,N-1)$, where w is an odd number called the chord length, and an additional edge (chord) between node i and node j if and only if*

- $j \equiv \begin{cases} i \ominus_N w & \text{if } i \bmod 2 \equiv 0 \\ i \oplus_N w & \text{if } i \bmod 2 \equiv 1 \end{cases}$

So the degree of this topology is constant $d = 3$, the **connection costs** are comparable small. It is on the one hand possible to calculate the **diameter** as a function of N and w, or to

[23] A cycle in a topology is spanning, if it contains all nodes of the topology. Two cycles are disjoint, if they have no links in common.

[24] A left link leads to a node in the left subtree (the dashed lines in figure 3.10 (b)), a right link to a node in the right subtree (grey lines).



(a) Chordal Ring (b) Double Chordal Ring (c) Extended Chordal Ring

Figure 3.11: Additional links on rings

calculate the maximum number of nodes and the corresponding optimum chord length for any given diameter on the other hand (see Section 5.2).

An increase in the number of nodes is done by adding nodes to the ring but all chords have to be changed, so we have an **extensibility** type of 4. A **routing algorithm** is given by Arden and Lee [Arde 81]. This algorithm calculates a path between any two nodes by determing a sequence of ring and chord traversals, i. e. using an ring edge or an additional edge for transmission. In order to determine this path, it is sufficient to know the source and destination address, N and w.

A Chordal Ring on 16 nodes is shown in figure 3.11 (a), with $w = 5$.

Extended Chordal Ring ECR(W) [Doty 82], [Doty 84] As we can see in the Chordal Ring there are two different connection patterns, one for all even nodes and one for all odd nodes. Doty has extended this model to p different patterns where p is called the period of the ring, where all nodes in the same residue class (mod$p$) have the same connection pattern. ECR's are furthermore defined for any degree $d > 2$.

Definition 36 *An Extended Chordal Ring, denoted by $ECR(W)$, is a ring of N nodes, where W is a $(p \times (d-2))$ chord matrix with elements $_{ij}$, and p, d are integers with $d > 2$, $1 \leq p \leq N/2$, $N \bmod p \equiv 0$, and $d-2$ additional edge between node x an d node y if and only if*

- *$x \bmod p \equiv i$ and $y \equiv x \oplus_N w_{ij}$ for $j = 1, 2, \ldots, d-2$.*

When constructing an ECR we first have to choose upon its degree d and its period p. When choosing the edges between pairs of nodes (assigning values to the elements of the chord matrix), we have to consider, that adding a chord w_{ij} to node i also adds a chord to node $i \oplus_N w_{ij}$, which results in an entry in the chord matrix in row $i \oplus_p w_{ij}$ with value $-w_{ij}$. So there are only $\frac{p(d-2)}{2}$ chords which we can choose freely, the others are fixed.

ECR's might serve very well as an interconnection network. They are regular and p-symmetric, optimally **fault tolerant** and the **connection costs** increase only linear with N.

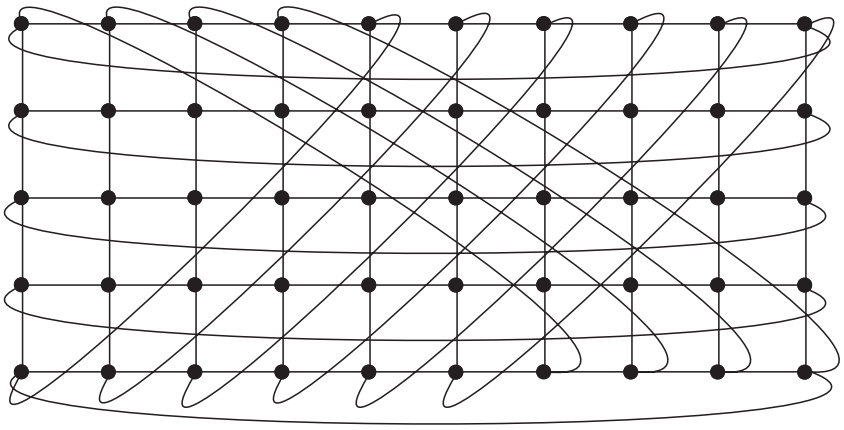


Figure 3.12: Additional links on a mesh: $MDMW(50)$

There exists a type 3 routing algorithm for Generalized Chordal Ring (see Section 3.6), that can be applied for ECRs. Furthermore ECR's possess small **communication delays** (see Chapter 5). [25]

Figure 3.11 (c) shows an Extended Chordal Ring with degree 4.

Double-Chordal Ring DCR($\mathbf{w_1}, \mathbf{w_2}$) [Jin 84]

Definition 37 *A Double-Chordal Ring, denoted by DCR(w_1,w_2) is an ECR with* $N \bmod 2 \equiv 0$ *nodes and* $W = \begin{pmatrix} w_1 & w_2 \\ -w_1 & -w_2 \end{pmatrix}$ *where w_1 and w_2 are odd numbers with* $w_2 < w_1 \leq N/2$.

Jin and Yang [Jin 84] have tried to find formulas for diameter analogous to Arden [Arde 81], but were less succesfull. They actually presented a closed form for calculating optimum chord lengths for a given diameter, but their results are wrong. [26]

Furthermore, there is no reason for restricting w_i to odd numbers. If we use even numbered chord lengths, we obtain DCRs with lower diameter and average distance (see Section 5.2).

Figure 3.11 (b) shows a Double Chordal Ring.

Minimum Distance Mesh with Wrap aroud Links MDMW(N) [Beiv 87a, Beiv 87b] In contrast to a torus, where the first and last nodes of the same rows and columns are connected, links between nodes in different columns are added in a MDMW in order to minimize diameter.

Definition 38 *In a Minimum Distance Mesh with Wrap around Links $MDMW(N)$ N nodes are arranged in a $(h \times v)$-grid* [27]*, and links are added according to the following rules:*

- *the lowest node in column j $1 \leq j \leq v$ is connected to the highest node in column* $(j + b - 1) \bmod v$ *for all rows (vertical wrap around links)*
- *the first node is connected to the last node in every row i $1 \leq i \leq h$ (horizontal wrap around links)*

[25] Nevertheless, this topology has not been used in any existing parallel computer, maybe because of less popularity. Even in a specific reconfigurable system, whose vendors claim that it is possible to configure *all* possible topologies, ECR's cannot be build.

[26] I have found a lot of contradictory examples to their results.

[27] If $N < hv$ then a leftmost upper rectangle of dimension $(r \times (v - b + 1))$ is discarded from the grid.

with

$$\begin{aligned} b &= \lceil\sqrt{N/2}\rceil, \\ r &= \lceil N/b\rceil b - N, \\ h &= b + r, \\ v &= \lceil N/b\rceil - r. \end{aligned}$$

Figure 3.12 shows a $MDMW$ of 50 nodes.

This topology is regular, and has only linearly growing **connection costs** (constant degree $d = 4$). The **diameter** is given by $k = b = \lceil\sqrt{N/2}\rceil$, which is a considerable improvement to an ordinary mesh. A $MDMW$ is **extensible** to all values of N, but it is necessary to rearrange most of the links.

A **routing** algorithm (implemented in OCCAM 2) is given in [Arru 90]. The basic idea is that a message is first sent b-times in horizontal direction and then $b-1$ times in vertical direction, where an appropriate labelling scheme enables to calculate, whether the message should be sent east or west in horizontal moves, and north or south in vertical moves respectively.

Banyan Hypercube BQ(h, n, s) [Yous 90] A Banyan network is a multistage interconnection network [Adam 87], its topology is a directed graph with a unique path from every base to every apex[28] where normally the bases and apexis represent processors or memory modules, whereas all other nodes are switching cells (see [Goke 73] or [Sieg 79] for a more detailed discussion of Banyan networks). In a regular Banyan network the indegree s (number of ingoing links) equals the outdegree, and there are s^n nodes within each level.

In a Banyan Hypercube all nodes in the Banyan network represent processing elements, connected via bidirectional links, and there are additional links between nodes in the same label forming a Boolean n-cube.

Definition 39 *A Banyan Hypercube, denoted by $BQ(n,s)$, consists of $(n+1)s^n$ nodes, n,s are integers, s is a power of two, $(l|x)$ denotes the address of node x (in a n-digit radix s representation) in level l $(0 \le l \le n)$.*
Node $(l|x)$ is connected to $(m|y)$ if and only if

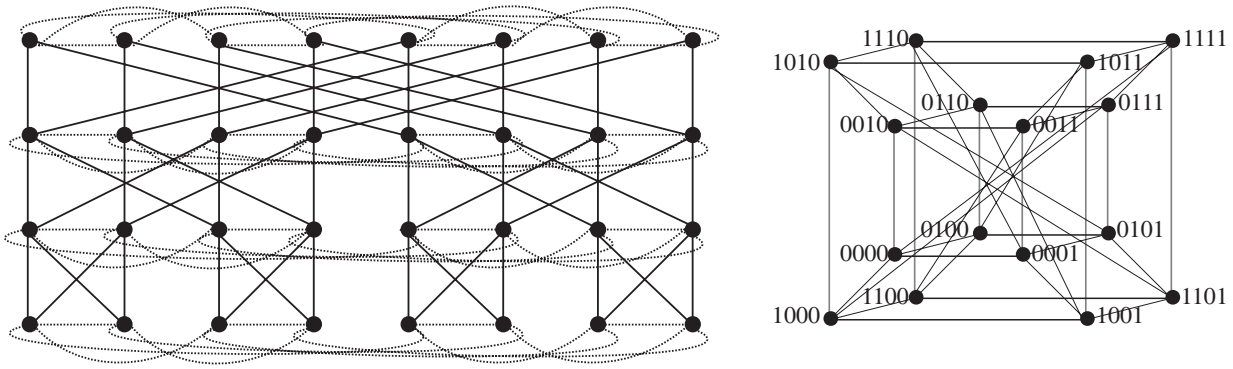
- *$l < n+1$ and $m = l+1$ and $y = x_{n-1}\cdots x_{l+1} a x_{l-1}\cdots x_0$ for $a = 0, 1, \ldots, s-1$.*
- *$l > 0$ and $m = l-1$ and $y = x_{n-1}\cdots x_l a x_{l-2}\cdots x_0$ for $a = 0, 1, \ldots, s-1$.*
- *$l = m$ and x and y differ exactly in one digit.*

When constructing a $BQ(n,s)$ one starts with arranging the $(n+1)s^n$ nodes in $n+1$ levels, each containing exactly s^n nodes. The nodes within each level are labelled by n-digit radix s numbers. All nodes (except those at the last level) are now connected to their s "successors" in the previous level, which are those nodes that have either the same address or differ only in the l-th digit (l is the level number). These connections form a rectangular banyan network (the black full lines in figure 3.13 show the banynan connections). Now the additional links are established among the nodes in the same level forming a Boolean n-cube (the dashed lines in figure 3.13 fig).

Since the nodes in the first and last level have less edges than the inner nodes, BQ's are irregular. The nodes in level 0 and n have a node degree of $s + n\log_2 s$, whereas all other nodes have $d_i = 2s + n\log_2 s$. **Diameter and average distance** have only a closed form for $s = 2$ and $s = 4$ (see [Yous 90] for a proof), and are shown in table 3.6. For other values of s the authors have not found a closed form.

BQ's have comparably high **connection costs**. A recursive construction rule can be given showing the **extensibility**:

[28] A node is a base, if there are no incident arcs into it, a node is an apex, if there are no incident arcs out of it.



(a) Banyan Hypercube (b) Folded Hypercube

Figure 3.13: Additional links on hypercubes: $BQ(4,2,3)$, $FQ(4)$

1. A $BQ(0,s)$ is a single node.
2. A $BQ(n,s)$ can be constructed from s copies of $BQ(n-1,s)$ by connecting the corresponding nodes.

So BQ's have inherited the hierarchical structure of Banyan Networks and hypercubes.

A **routing algorithm** is given by Youssef and Narahari in their article. The message is first sent to the destination level using the banyan edges, and then to the destination itself using n-cube edges. We have already presented the routing algorithm for n-cubes (section 3.3), and must now specify which of the banyan edges are chosen when changing levels. If the destination level l' is above the current level l, the message is routed to this node, which has the same digit at the l-th position as the destination address. If the destination level l' is beneath the current level l, the message is routed to this node, which has the same digit at the $(l-1)$-th position as the destination address. This guarantees, that a minimum number of n-cube routing steps will be necassary.

Folded Hypercubes FQ(n) [El A 91]

Definition 40 *A Folded Hypercube, denoted by $FQ(n)$, of dimension n is a Boolean n-cube with an additional link between node u and v if and only if*

- *their binary addresses differ in all bits.*

The additional links are shown in figure 3.13 as dashed light lines.

An FQ is symmetric and regular of degree $n+1$, the **connection costs** are somewhat higher than in a $Q(n)$. The reduction in **communication delays** might justify the comparable small increase in degree. However if the number of links per PE is restricted by hardware, the size of the largest possible FQ is only half of the size of the largest possible Boolean n-cube.

The **routing** is similar to routing in hypercubes: a node compares its address with the destination address using a binary "xor" operation. If the number of 1's in the result is greater than $n/2$ the message is routed via the additional link otherwise ordinary hypercube routing takes place.

The hierarchical structure of hypercubes is lost in a FQ, i. e. if we want to **extend** the structure to higher dimensions, it is necessary to rearrange the additional links.

	N	d	L	k
$BQ(n,s)$	$(n+1)s^n$	$2s + n\log_2 s$		n if $s=2$ $2n$ if $s=4$
$CR(w)$	N	3	$3N/2$	
$DCR(w_1,w_2)$	N	d	$Nd/2$	
$ECR(W)$	N	4	$2N$	
$FQ(n)$	2^n	$n+1$	$(n+1)2^{n-1}$	$\lceil n/2 \rceil$
$HT(h)$	$2^h - 1$	4	$3(2^{h-1}-1)$	$\lfloor 3(h-1)/2 \rfloor$
$MDMW(N)$	N	4	$2N$	$\lceil\sqrt{N/2}\rceil - 1$ if $N \le 2b^2 - 2b + 1$ $\lceil\sqrt{N/2}\rceil$ otherwise
$ST(h)$	$2^{h+1}-1$	4	$2N$	

	Average Distance	symmetry	extensibility type	routing type
$BQ(n,s)$	$n/2 + \frac{(n+1)^2-1}{6(n+1)}$ if $s=2$ n if $s=2n$	n	2 or 6	2
$CR(w)$		2	3	1
$DCR(W)$		p	3	3
$ECR(w_1,w_2)$		2	3	1
$FQ(n)$	$\frac{\sum_{i=1}^{k} i Z_i}{N-1}$ $Z_i = \begin{cases} \binom{n+1}{i} & \text{if } i < k \\ \binom{n+1}{n/2+1} & \text{if } i = k \text{ and } n \text{ even} \\ \binom{n}{i} & \text{if } i = k \text{ and } n \text{ even} \end{cases}$	1	8	1
$HT(h)$	$\frac{5(h-1)}{4} - \frac{4}{3} + \frac{4}{2^{h-1}3} - \frac{(h-1) \bmod 2}{12}$	$h+1$	1	1
$MDMW(N)$	$k(1 - \frac{2(k^2-1)}{3(N-1)})$	1	3	2
$ST(h)$				

Table 3.6: Additional Links

Among the structures presented in this section, ECR's and hypertrees seems the most important ones for interconnection networks. Both are regular and offer a high degree of symmetry (see tabel 3.6). The hypertree has the advantage, that it is easier to extend (because of the hierarchical tree structure), but has higher communication delays. ECR's have lower diameter, but the disadvantage, that it is not possible to determine the optimum chord lengths, except by exhaustive search.

Folded Hypercubes and Banyan Hypercubes suffer from their high connection costs, which make them inattractive for large systems.

3.6 Generalized Chordal Rings

Generalized Chordal Rings [Berm 86] are based on the idea of Chordal Rings proposed by Arden and Lee, but weakening the restriction that all nodes have to lie on a ring.

Definition 41 *A Generalized Chordal Ring, denoted by $GCR(W)$, consists of N nodes (numbered from 0 to $N-1$), p is a divisor of N called the period of the GCR, W is the chord matrix with elements w_{ij}. Node i is connected to node j if and only if*

- *node $i \oplus_N p$ is connected to $j \oplus_N p$.*

The nodes are divided into p residue classes and all chord lengths are kept in a $(p \times d)$ chord matrix W. The value of p corresponds to the symmetry of the topology (GCR's are p-symmetric) and the set of chord lengths determines the diameter.

A Generalized Chordal Ring is constructed as follows:

1. Find an appropriate period p with $1 \leq p \leq N$ and $N \bmod p \equiv 0$.
2. For every period, find values w_{ij} $0 \leq i \leq p-1$, $1 \leq j \leq d$, with $0 \leq w_{ij} \leq N-1$, so that $d_i \leq d$ for any node i.[29]

For a given number of nodes, there exists a great variety of GCRs, which show great variations in diameter, average distance and other measures. The construction neither guarantees that the GCR is regular, nor that it is connected.

Arden [Arde 90] has developed a recursive **routing algorithm** for point-to-point communication in Generalized Chordal Rings. In a first step (which must be executed only once) "distance tables" are established for the nodes $0, 1, \ldots, p-1$ (as representatives of their residue classes). A node appears in level i in the distance table of node j, if it is at distance $\leq 2^i$ from the representative j in this residue class. [30] The tables for all other nodes in the same residue class are obtained easily by adding the node label to all node addresses (modulo N) in the original table of the corresponding period. When routing a message, the path search starts at the lowest level l. If the source and destination node are root and leaf in the same table, the recursion is continued at the next level $l-1$. Otherwise a common node [31] is searched at level l in the source table and at level $l-1$ of the destination table. If this search fails, level l in the destination table is investigated.[32] The problem is then split into two subparts, first searching a path from the source to the common node, second searching a path from the common node to the destination, both at the next higher level ($l \rightarrow l-1$).

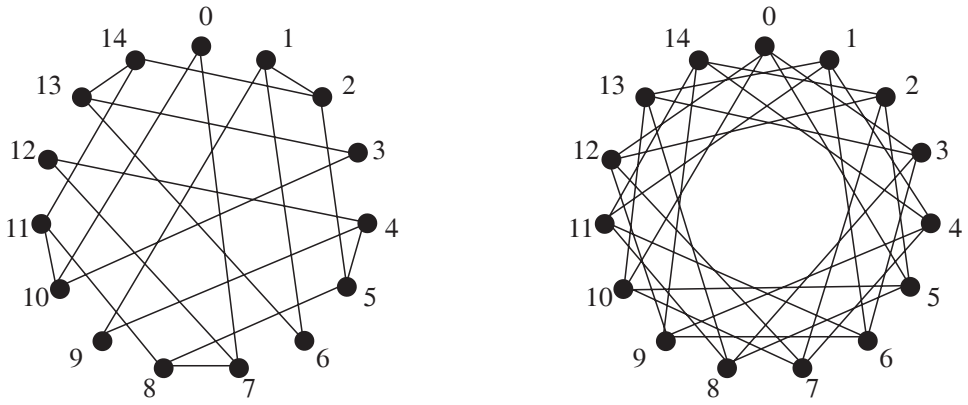
A special type of Generalized Chordal Rings are **Circulant graphs** [Boes 85] with period $p = 1$, so the chord matrix consists of a single row, here called *jump sequence*.

[29] Note that only $pd/2$ chord lengths has to be found since the allocation of any w to node i fixes a chord length of node $i \oplus_N w$ to $-w$.

[30] The tables are redundant half-trees, and can be stored efficiently (using less space) in a bit vector representation.

[31] A common node is a node which appears in both tables.

[32] This search must be successfull, because the maximum distance between any two nodes is not larger than 2^l.



(a) Generalized Chordal Ring (b) Circulant

Figure 3.14: Generalized Chordal Rings: $GCR(0,7,-5;1,5,-7;-1,3,-3)$, $CG(3,5)$

	N	d	L	k	symmetry	extensibility	routing
GCR	N	d	$dN/2$		p	3	3
CG$(1,n_2)$	N	4	$2N$	$\lceil\frac{\sqrt{2N-1}-1}{2}\rceil$	1	3	3

Table 3.7: Generalized Chordal Rings

Definition 42 *A Circulant graph denoted by $CG(n_1,n_2,\ldots,n_j)$ comprises N nodes, where $n_1,n_2,\ldots,n_j$ are integers with $0<n_1<\cdots<n_j<(N+1)/2$ called the jump sequence, such that node x is connected to node y if and only if*

- $y \oplus_N n_i \equiv x$ *or*
- $y \ominus_N n_i \equiv x$.

Circulant graphs are regular of degree $2j$, values for diameter vary strongly for different jump sequences. In [Boes 85] a lower bound for the diameter in circulant graphs is given but the problem of determining that jump sequence reaching this bound has so far only been solved for $d=4$ circulant graphs. In this subclass the CG$(1,\lceil\sqrt{2N-1}\rceil)$[33] is optimum with respect to diameter ($k=\lceil\sqrt{2N-1}\rceil$). This optimum degree 4 circulant will be used in table 3.7 for comparison.

3.7 Combination of Basic Modules

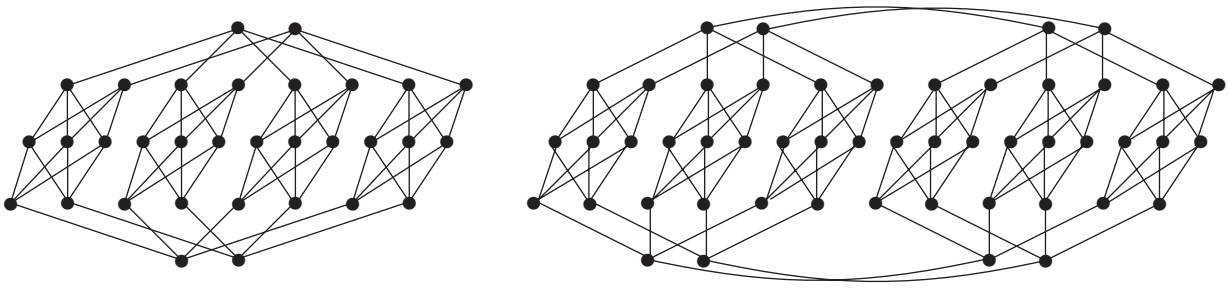
The topologies dealt within this chapter are constructed by connecting graphs (basic modules) to a higher level graph. The connection can either be done by merging nodes or by linking them together. Both methods causes an increase in the degree of the nodes used for connection. It is advantageous to use irregular structures as basic modules and connect them by using the $d-d_i$ "free" links of nodes with smaller node degree in order to make them regular. That is why trees are frequently used as basic modules.

3.7.1 Merging Methods

F-tree FT$(\mathbf{B},\mathbf{h})$ Friedman [Frie 66] [34]

[33] Note that this Circulant is also an ECR.

[34] In this article no explicit name for the topology was given so I decided to call it F-tree.



(a) F-tree (b) K-tree

Figure 3.15: Merging of trees: $FT(3,2)$, $KT(3,2)$

Definition 43 *An F-tree, denoted by $FT(B,h)$, combines $B+1$ copies of a $T^{max}(B,h)$ (component trees) by merging their leaf nodes.*

A maximum tree $T^{max}(B,h)$ is of degree $d = B+1$, so B edges can be added to every leaf node without increasing the degree of the tree and it is therefore possible to merge $B+1$ maximum trees, resulting in a regular graph with degree $d = B+1$. The total number of nodes is given by

$$N = \frac{2(B+1)(B^h - 1)}{B-1}$$

The **diameter** of a FT is given by $k = 2h$, i. e. twice the height of the component tree.

There are d redundant paths between any pair of nodes so the **fault tolerance** is improved significantly compared to ordinary trees. But the problem of high traffic density near the root nodes still exists.

The structure can either be **extended** by increasing the number of branches or levels. The first possibility leads to an increase in degree whereas the latter increases diameter. Both methods require restructuring of the existing F-tree and it is not possible to add an arbitrary number of nodes.

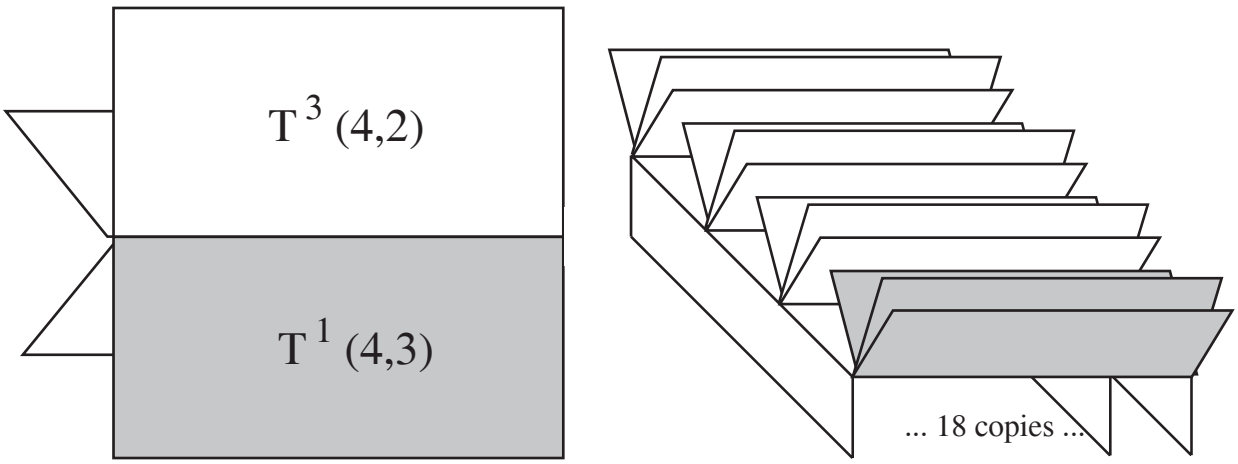
No **routing algorithm** is given in [Frie 66] for F-trees, but since they are tree based structures, an algorithm can be derived easily if the nodes are identified by their position in the component tree and by the number of the component tree. A node receiving a message has to decide whether the destination node is in the same component tree or not. If so, then ordinary tree routing takes place (see Section 3.1). Otherwise the message has to be sent downwards to the leaves and then to the destination component tree.

K-tree KT(B, h) [Korn 67] K-trees are based on the idea of Friedman with the only difference that maximum 2-rooted trees are used.

Definition 44 *A K-tree, denoted by $KT(B,h)$, combines $B+1$ copies of a $T^{2,max}(B,h)$ by merging their leaf nodes.*

The total number of nodes is given by

$$N = \frac{2(2B^{h+1} - (B+1)}{B-1}$$



(a) Nonidentical Hinging Graph (b) Nonidentical Blocking Graph

Figure 3.16: Merging of trees: $NHG(4,3,3,1)$, $NBG(4,3,3,1)$

with degree $B+1$ and diameter $k=2h+1$. **Routing** and **extensibility** are analogous to F-trees. The only difference of the two structures is that F-trees always have even diameter, whereas the diameter in K-trees is always odd.

Nonidentical Hinging and Blocking graph N.G($\mathbf{d,l,m_1,m_2}$) [Stor 70]

Definition 45 *A Nonidentical Hinging graph, denoted by $NHG(d,l,m_1,m_2)$, combines $d-1$ copies of a T^{m_1} and one T^{m_2} by merging their leaf nodes.*

A Nonidentical Blocking graph $NBG(m_1,m_2,l,d)$ combines $m_1(d-m_1+1)(d-1)^{l-2}$ subgraphs of type T_{m1} in a way that their leaf nodes are the columns of a grid. The nodes in the first row are also the leaves of $d-1$ hinged T_{m_2} subgraphs, and all other rows are also the leaves of $d-1$ hinged T_{m_1} subgraphs.

T^{m_1} denotes a m_1 rooted maximum tree with l levels (height $l-1$), labelled from $(0,1,\ldots,l-1)$ and degree d with $l>1$ and $2\leq m_1\leq d-1$.
T^{m_2} denotes a m_2 rooted subtree with $l+1$ levels and $1\leq m_2\leq d-1$; all leaf nodes have a node degree of 1; all nodes in level $l-1$ have a node degree $2\leq d_i\leq d$, and for all other nodes $d_i=d$.
The number of leaves in T_{m_2} equals that in T_{m_1}.

Figure 3.16 illustrates the construction of an NHG and an NBG. All white planes represent the first component tree T^{m_1}, each grey planes stand for a copy of T^{m_2}.

Both structures are irregular since some of the nodes in the l-th level of T^{m_2} must have a node degree less than d in order to fulfill the requirement that both subtrees have the same number of leaf nodes which is guaranteed by the following equation:

$$m_2(d-m_2+1)\leq m_1(d-m_1+1)\leq m_2(d-m_2+1)(d-1)$$

Extensibility is of type 4 if the number of levels is increased or even worse of type 8 if the degree is increased. Values for diameter and N can be given in a closed form of d,l,m_1 and m_2:

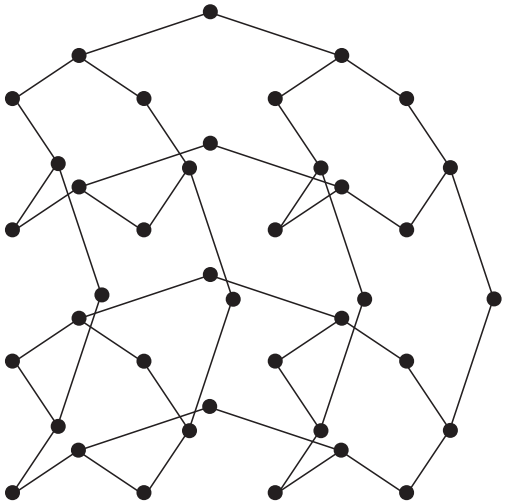


Figure 3.17: Merging of trees: $OT(2)$

$$
\begin{aligned}
N_{NHG} &= \frac{1}{d-2}(m_1(m_1-3)(d-1)+m_2(m_2-3) \\
&\quad +(d-1)^{l-2}(m_1(2d-3)(d-m_1+1)+m_2(d-1)(d-m_2+1)))
\end{aligned}
$$

$$
k_{NHG} = \begin{cases} 2l-1, & m_1=1, \quad m_2=1 \\ 2l-1, & m_1=1, \quad m_2>1 \\ 2l-1, & m_1>1, \quad m_2=1 \\ 2l, & m_1>1, \quad m_2>1 \end{cases}
$$

$$
\begin{aligned}
N_{NBG} &= \frac{m_1^2(d-m_1+1)(d-1)^{l-2}}{d-2}(d(m_1-3)+2(d-m_1+1)(d-1)^{h-1}) \\
&\quad +\frac{m_2(d-1)}{d-2}(m_2-3+(d-m_2+1)(d-1)^{l-1}) \\
&\quad -\frac{m_1(d-1)}{d-2}(m_1-3+(d-m_1+1)(d-1)^{l-2})
\end{aligned}
$$

$$
k_{NBG} = \begin{cases} 4l-3, & m_1=1, \quad m_2=1 \\ 4l-3, & m_1=1, \quad m_2>1 \\ 4l-3, & m_1>1, \quad m_2=1 \\ 4l-1, & m_1>1, \quad m_2>1 \end{cases}
$$

In Section 5.2 a table with maximum NHG's and NBG's will be given.

I could not find a **routing algorithm** for an NHG or NBG in the literature.

Orthogonal tree network OT(h) [Nath 83] An OTN consists of binary trees, where every tree is connected to some others via a specific common leaf node.

	N	d	L	k
$FT(B,h)$	$\frac{2(B+1)(B^h-1)}{B-1}$	$B+1$	$(B+1)N/2$	$2h$
$KT(B,h)$	$\frac{2(2B^{h+1}-(B+1))}{B-1}$	$B+1$	$(B+1)N/2$	$2h+1$
$NBG(d,l,m_1,m_2)$		$2 \leq d_i \leq d$		
$NHG(d,l,m_1,m_2)$		$2 \leq d_i \leq d$	$((d-1)L_{S_1}+L_{S_2})/2$	
$OT(h)$	$2^{2h}+2^{h+1}(2^h-1)$	$2 \leq d_i \leq 3$	$2^h(2^{h+1}+1)$	$4h$

	symmetry	extensibility type	routing type
$FT(B,h)$	$h+1$	4 or 8	4
$KT(B,h)$	$h+1$	4 or 8	4
$NBG(d,l,m_1,m_2)$	$2h$	4 or 8	4
$NHG(d,l,m_1,m_2)$	$2h$	4 or 8	4
$OT(h)$	h	2	4

Table 3.8: Merging Methods

Definition 46 *An Orthogonal tree network, denoted by $OTN(h)$, is a grid of $2^h \times 2^h$ (base) nodes and $2^{h+1}(2^h-1)$ (internal) nodes, where the 2^{h-1} nodes in every row and column are the leaf nodes of a binary tree of height h.*

The total number of nodes in an $OTN(h)$ is given by $N = 2^{2h} + 2^{h+1}(2^h - 1)$, arranged in 2^{h+1} binary trees, where each leave is shared by two trees. The roots and the leaves have a node degree of 2, the inner nodes have a node degree of 3, so the topology is irregular.

An $OTN(h)$ is $h+1$-symmetric and a hierarchic structure since 4 $OTN(h)$ can be combined to an $OTN(h+1)$ by linking corresponding old root nodes via new root nodes as shown in figure 3.17.

The constructions of Friedman and Korn result in regular structures, whereas all others are irregular (compare table 3.9). The OTN has the lowest connection costs, the hinging and blocking graphs have smallest **communication delays** out of this group of graphs.

I have not found **routing** algorithms for those structures in literature, but all structures are based on trees, so it should be possible to derive algorithms based on tree routing. The **extensibility** for hinging and blocking graphs is of type 4, if a new level is added, and type 8 if the number of branches is increased. Only the OTN has a better extensibility, because of its hierarchical structure.

3.7.2 Linking Methods

Hypernet Hnet(B, e, G, h) [Hwan 87] Hypernets are hierarchical compositions of basic modules characterized by 4 parameters: the set of basic modules B, the number of links leading to other basic modules (external links) 2^e, the number of links between any two basic modules (global connectivity G) and the number of hierarchy levels h. Any arbitrary structure can be used as basic module, with the only restriction that every node in the basic module must have at least one external link so the number of nodes in the basic module is bounded by 2^e.

Definition 47 *A Hypernet, denoted by $Hnet(B,e,G,1)$, of level 1 is a basic module with 2^e external links. A hypernet of level h $Hnet(B,e,G,h)$ connects $2^{e-1}/G$ nets of level $h-1$ such that there are exactly G links between any two subnets.*

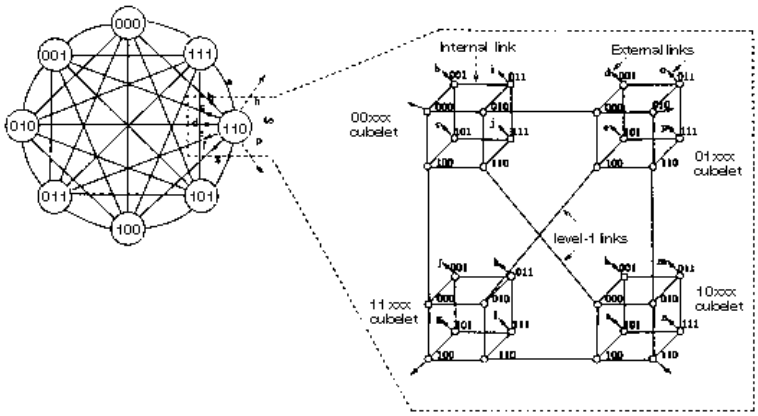


Figure 3.18: Linking of cubes: $Hnet(Q_3, 3, 3, 1)$

The construction will be demonstrated for a level 3 hypernet, based on a Boolean 3-cube with $G = 1$, $Hnet(Q(3), 3, 3, 1)$ (see figure 3.18, taken from [Hwan 87]). The construction starts with a level 1 net, a $Q(3)$. Then, $2^{d-1}/G$ basic modules are combined to a level 2 net, so we need 4 cubes, each connected with G links to every other (in our example, $G = 1$, there is a single link between every pair of $Q(3)$s). For the next level, we need 8 level-2 nets, which are shown as circles, and again establish G links between every pair of subnets. Whenever $G = 1$, the higher level net will be a complete graph with the subnets as nodes.

Values for **diameter** and average distance depend on the types of basic modules. If identical basic modules are used the diameter is given by

$$k = 2^{h-1}(k_1 + 1) - 1$$

and the average distance is given by

$$\mu = 2^{h-1}(\mu_1 + 1) - 1$$

where k_1 denotes the diameter and μ_1 the average distance in the basic module.

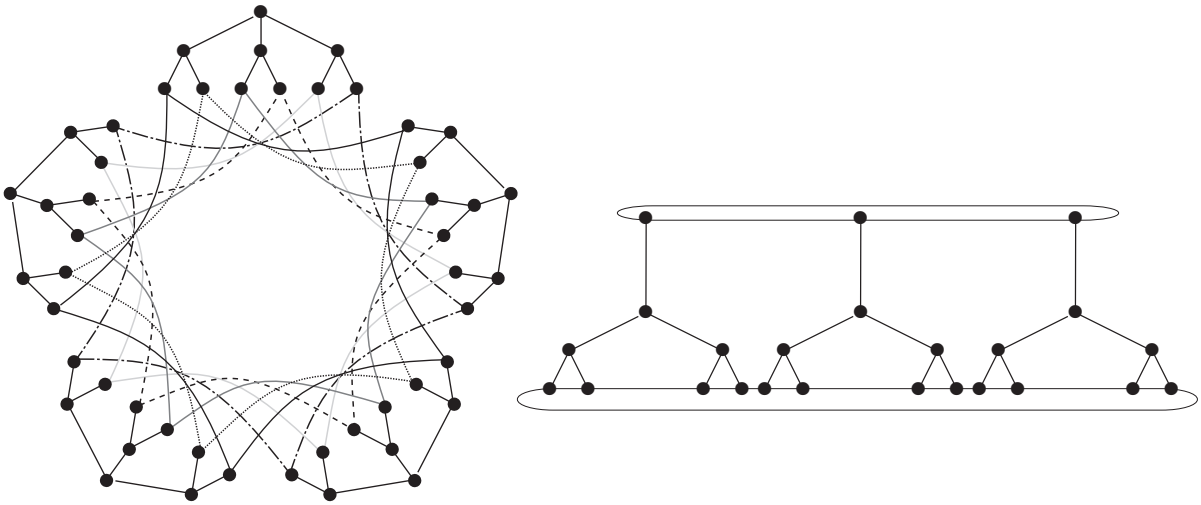
The hierarchical structure of hypernets allows to increase the size of the network without restructuring the existing parts. So **extensibility** is of type 2.

In [Hwan 87] a **routing** algorithm is given.

$\mathcal{D}$-tree $\mathcal{DT}^m(B, h, K, p)$ [Ramk 88] A $\mathcal{D}$-tree consists of single-rooted (m=1) or two-rooted (m=2) maximum trees[35] of arbitrary degree and height connected via their leaves.

Definition 48 *A $\mathcal{D}_K$-tree denoted by $\mathcal{D}T^m(B, h, K, p)$ contains p copies of $T^{m,max}(B, h)$ and all leaves with label i are connected as a regular graph of degree B of order p and diameter K, where the leaves in a subtree are labelled from 1 to 2^{h-1}.*

[35] We will denote a m rooted maximum tree by $rootedmaximumT^{m,max}$. Such a tree has m root nodes and all non leaf nodes are of degree $B + 1$.



(a) D-tree (b) Multitree

Figure 3.19: Linking of trees: $\mathcal{D}T^1(2,2,2,5)$, $MTS(3,3,3)$

Figure 3.19 shows a single rooted $\mathcal{D}$-tree of degree 3, the graph used for leaf interconnection is a ring of 5 nodes. In Section 5.2 a detailed construction algorithm is given, trying to find a $\mathcal{D}T$ with minimum diameter.

$\mathcal{D}$-trees are regular of degree $d = B + 1$. **Communication delays** depend on the graph chosen for leaf interconnection, so these should be chosen to be optimum with respect to diameter, in order minimize the diameter of the complete structure. In a $\mathcal{D}_1$-tree, the graph used for leaf interconnection must be complete, and therefore $p = d$.

The total number of nodes in a $\mathcal{D}_K$-tree is given by

$$\begin{aligned} N &= \frac{pd(d-1)^h - 1}{d-2} && \text{if } m = 1 \\ &= \frac{2p(d-1)^{h+1} - 1}{d-2} && \text{if } m = 2 \end{aligned}$$

$\mathcal{D}_K$-trees are regular, h-symmetric and their diameter is given by

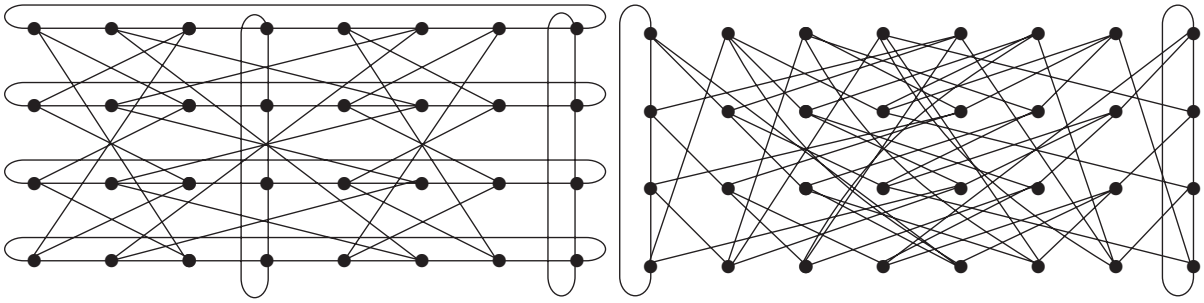
$$\begin{aligned} k &= (2h + K) && \text{if } m = 1 \\ &= (2h + K + 1) && \text{if } m = 2 \end{aligned}$$

Although $\mathcal{D}_K$-trees have optimum **fault tolerance** measured in cohesion, the failure of a link causes a significant increase in diameter so their persistence is poor.

Multitree Structured graphs MTS(m, d, h) [Arde 78]

Definition 49 *A Multitree Strucutred graph, denoted by $MTS(m,d,h)$, consists of m copies of $T1$[36] (also called component trees), whose root nodes are arranged in a cycle, where each leafnode is connected to $d-1$ other leafnodes forming at least one cycle.*

[36] $T1$ denotes a subtree with h levels, a root node with node degree 1, and all other nonleaf nodes with node degree d.



(a) Torus

(b) C1 Graph

Figure 3.20: Linking of rings: $Tor(4,8,4,7)$, $C1(4,4)$

Figure 3.19 (b) shows a multitree, made up of 3 subtrees of height 3 and degree 3.

The total number of nodes is given by

$$N = m(1 + \frac{(d-1)^{h-1} - 1}{d-1})$$

values for **diameter** depend on the chosen leaf connection. It has neither been possible to define connection rule, which would result in a graph of minimum diameter, nor to give a closed form for diameter as a function of m, d and h.

No **routing algorithm** is given by the authors.

Torus Tor($\mathbf{m}, \mathbf{r}, \mathbf{p_{up}}, \mathbf{p_{down}}$) [Cont 83]

Definition 50 *A torus, denoted by $Tor(m, r, p_{up}, p_{down})$, consists of rm nodes, m denotes the number of nodes in a ring; r denotes the number of rings; p_{up}, p_{down} are integers relative prime to m; $(i|j)$ denotes the node address with 0 leqi $< r$ and $0 < j \leq m$. Node (i,j) is connected to node (k,l) if and only if*

- *$i = k$ and $j \equiv l \oplus_m 1$ or $j \equiv l \ominus_m 1$ (connections within the ring) or*
- *$k \equiv i \oplus_m 1$ and $l \equiv i \cdot p_{up} \bmod m$ or*
- *$k \equiv i \ominus_m 1$ and $l \equiv i \cdot p_{down} \bmod m$.*

A torus consists of r cycles, each containing m nodes. Each node has a node degree of 4, two of its edges are used for connection within the ring, one is used for the upward, the other for the downward connection.

Figure 3.20 (a) shall illustrate the construction of a $Tor(4,8,4,7)$. First, $m = 4$ rings, each containing $r = 8$ nodes are arranged above each other. Then, either the upward or the downward connections are established (each upward link corresponds to a downward link), connecting the nodes in one level to the nodes in the neighbouring levels (the first and the last level are said to be neighbours, too).

A torus is regular of degree 4 (this is guaranteed, since m and p_{up} as well as m and p_{down} are relative prime), values for **diameter** and **average distance** depend on the values chosen for p_{up} and p_{down}, some good results will be shown in Section 5.2.

If $p_{up} = p_{down} = 1$ then the structure is equivalent to a wrap around mesh, if $r = 1$ it is an Extended Chordal Ring (definition 36).

	N	d	L	k
$\mathcal{DT}^1(B,h,p,K)$	$p\frac{(B+1)B^h-2}{B-1}$	$B+1$	$Nd/2$	$2h+K$
$Hnet(Q(3),3,3,h)$	$2^{2^{h-1}+h+1}$	4		$2^{h+1}-1$
$MTS(m,B,h)$	$m(1+\frac{B^{h-1}}{B-1})$	$B+1$	$\frac{N(B+1)}{2}$	
$C1(d,m)$	$m(d/2)^{m-1}$	d	md^m2^{m-2}	m for $m=3$ $m+\lceil\frac{m-1}{2}\rceil-2$ otherwise
$Tor(r,m,p_{up},p_{down})$	rm	4	$rmd/2$	

	symmetry	extensibility type	routing type
$\mathcal{DT}^n(B,h,p,K)$	h	4,7 or 8	3
$Hnet(Q(3),3,3,1)$	1	1	1
$MTS(m,d,h)$	$h-1+2^{h-2}$	varies	
$C1(d,m)$	node	4 or 8	
$Tor(r,m,p_{up},p_{down})$		4	

Table 3.9: Linking methods

A modification can be made in order to obtain a degree of 3 by replacing every node in a torus by a pair of nodes, where one edge connects the pair, the two remaining edges of one node are used for the connections within the level, and the edges of the other are used for the connections to other levels.

C1 Graph C1(d, m) [Memm 82]

Definition 51 *A C1 Graph, denoted by $C1(d,m)$, consists of mr nodes, $d \bmod 2 \equiv 0$, $m \geq 3$ denotes the number of nodes in a ring, $r = (d/2)^{m-1}$ is the number of rings, $(i|j)$ denotes the node address with $0 \leq i < r$ and $0 < j \leq m$. Node $(i|j)$ is connected to $(k|l)$ if and only if*

- $k \equiv i \oplus_m 1$
- $l \equiv x \oplus_r \frac{d}{2}(j-1)$ *for* $x = 1, 2, \ldots, d/2$.

In a $C1$ graph the nodes are arranged in the same way as in a torus, but the connection rule is different. Instead of connecting the nodes in the same level, all d links are used for connections to nodes in the neighbour levels. Furthermore, all even numbers are allowed for d, but it is not possible to choose the parameters for connection (as in Torus networks).

This has the advantage, that it is possible, to give a closed form for **diameter**:

$$k = \begin{cases} m & \text{for } m = 3 \\ m + \lceil\frac{m-1}{2}\rceil - 2 & \text{for } m > 3 \end{cases}$$

All structures presented in this section are regular, and with the exception of torus networks, where the degree is either 3 or 4, and the cube based hypernet ($d = 4$), the degree can be chosen freely (see table 3.9).

Hypernets are suitable for very large networks, because of their hierarchical construction. Torus networks should better be used for smaller systems ($N < 200$), because on the one hand it becomes computationally infeasible to try out all possible combinations of the parameters (m, r, p_{up}, and p_{down}), in order to obtain a network with the desired properties (e. g. small diameter). On the other hand von Conta ([Cont 83]) conjectures, that torus networks will not perform well with respect to communication delays, when N becomes large.

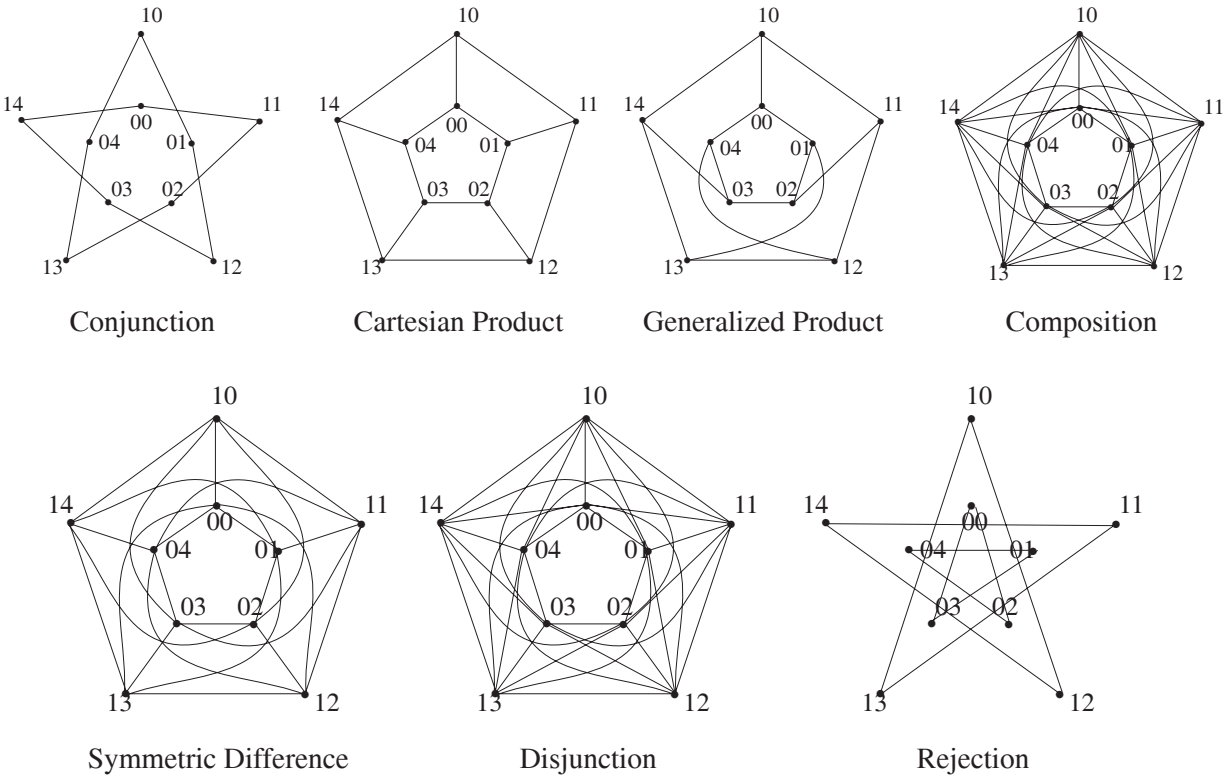


Figure 3.21: Boolean operations with $G_1 = K_2$ and $G_2 = C_5$

$\mathcal{D}$-trees, Torus entworks and $C1$ graphs possess comparably low communication delays, there will be further discussion in Section 5.2.

The multi-tree might also be an interesting structure, but we know only a few of its properties. Neither a routing algorithm, nor explicit construction rules nor values for diameter are known.

3.8 Boolean Operations on Graphs

Now we want to investigate topologies which are the result of a boolean operation on two graphs. In general a boolean operation $G_1 \circ G_2$ results in a graph G with $V = V_1 \times V_2$, the cartesian product of the node sets from graph G_1 and graph G_2. A node in G is labelled $(u_1|u_2)$ with $u_i \in V_i$ and the set of edges E of G is expressed in terms of the edges in E_1 and E_2, depending on the boolean operation applied.

Conjunction $\mathbf{G_1 \wedge G_2}$ [Weic 63][Ore 62] The conjunction of two graphs is perhaps the simplest operation. Two nodes $(u_1|u_2)$ and $(v_1|v_2)$ in G are connected only if (u_1, v_2) is and edge of G_1 **and** (u_2, v_2) is an edge ofG_2. It can be viewed as the "and-connection" of two graphs.

Figure 3.21 (a) shows the conjunction of a line and a cycle on 5 nodes.

In [McAn 63] and [Brua 67] the conjunction is modified and applied on directed graph.

Disjunction $\mathbf{G_1 \vee G_2}$ [Hara 67] The disjunction can be viewed as the "or-connection" of two graphs. Two nodes $(u_1|u_2)$ and $(v_1|v_2)$ are connected if (u_1, v_2) is and edge of G_1 **or** (u_2, v_2) is an edge of G_2.

Symmetric difference $G_1 \diamond G_2$ [Hara 67] The symmetric difference of two graphs is constructed by the "exclusive-or-conenction". Two nodes $(u_1|u_2)$ and $(v_1|v_2)$ are connected if either (u_1, v_2) is an edge of G_1 or (u_2, v_2) is an edge of G_2 (bot not both).

Rejection $G_1 \setminus G_2$ [Hara 67] The rejection can be viewed as the "nor-connection" of two graphs. Two nodes $(u_1|u_2)$ and $(v_1|v_2)$ are connected if neither (u_1, v_2) is an edge of G_1 nor (u_2, v_2) is an edge of G_2.

Cartesian Product $G_1 \times G_2$ [Sabi 60], [Catt 79] The cartesian product can be viewed as the "sum" of two graphs. Two nodes $(u_1|u_2)$ and $(v_1|v_2)$ are connected if either u_1 and v_1 are identical and (u_2, v_2) is an edge in G_2 or if the second components are identical $(u_2 = v_2)$ and the first components are an edge of G_1.

Some structures that we already know, can be described by the cartesian product. The Boolean n-cube is the cartesian product of a line and a Boolean $n-1$-cube $Q(1) = K_2$ and $Q(n) = K_2 \times Q(n-1)$; a torus (wrap around mesh) can be seen as the cartesian product of two cycles $WM(b,h) = C_b \times C_h$.

In figure 3.21 the cartesian product of a line and a ring is shown.

Generalized Product $G_1 \star G_2$ [Berm 84] In the Generalized Product G_1 must be a directed graph with E_1^{dir} denoting the sets of arcs. For each arc $(u_1 \mapsto v_1) \in E_1^{dir}$ it is necessary to define a function $f_{(u_1 \mapsto v_1)}(u_2)$, a "self-bijective" (mapping) function assigning each node $u_2 \in E_2$ to a node $f(u_2) \in E_2$. [37]

$G_1 \star G_2$ can be viewed as formed by $\mid V_1 \mid$ copies of G_2 where two copies generated by the nodes u_1 and v_1 are joined if $(u_1 \mapsto v_1)$ is an arc of G_1 according to the mapping function.

Bermond [Berm 84] has investigated the generalized product of a line (with orientation $(1 \mapsto 2)$) and a cycle: $G = K_2 \star C_{2a^2+2a+1}$ with $a \geq 1$ and $f_{(1 \mapsto 2)}(u_2) = (2a+1) \cdot u_2 \bmod n$. This graph is regular of degree 3 and it is possible to calculate its diameter:

$$k = \begin{cases} 2 & \text{if } a = 1 \\ a+2 & \text{if } a > 1 \end{cases}$$

If $a = 1$ this construction produces the famous Petersen graph (see figure 3.21 (b) and Chapter 5).

An extension allows the use of any directed graph G_1 with degree d_1, diameter k_1 and a given orientation E_1^{dir} instead of K_2, yielding a graph $G_1 \star C_{2a^2+2a+1}$ and $f_{(1 \mapsto 2)}(u_2) = (2a+1)u_2 \bmod n$. This graph is of degree $d = d_1 + 2$ and diameter $k_1 + a$.

Composition $G_1[G_2]$ [Hara 59], [Gome 85] Two nodes $(u_1|u_2)$ and $(v_1|v_2)$ are connected if either (u_1, v_1) is an edge in G_1 or if the first components are identical $(u_1 = v_1)$ and the second components are an edge of G_2. The composition of a line and a ring is shown in figure 3.21.

Table 3.10 summarizes the notations and construction rules for the boolean operations. It is important to note that boolean operations are in general not commutative, i. e. $G_1 \circ G_2 \neq G_2 \circ G_1$.

Most of the boolean operations can be described by operations on the adjacency matrices[38] of G_1 and G_2. In the last column of table 3.10 it is shown how the adjacency matrix of $G = G_1 \circ G_2$ is calculated.[39]

[37] If $f_{(u_1,v_1)}(u_2) = u_2$ for all arcs $(u_1; v_1)$ then the generalized product is simply the cartesian product.

[38] The adjacency matrix of $A = A(G) = [a_{ij}]$ of a graph G is the boolean $N \times N$ matrix with

$$a_{ij} = \begin{cases} 1 & \text{if there is an edge between } u_i \text{ and } u_j \\ 0 & \text{otherwise} \end{cases}$$

[39] I denotes the $N \times N$ identity matrix, J denotes an $N \times N$ matrix containing only 1's. $A * B$ denotes the tensor (or Kronecker) product. $A \oplus B = [a_{ij} \oplus_2 b_{ij}]$ denotes the addition (modulo 2) of A and B. ($\overline{A} = A \oplus I \oplus J$) denotes the complement of a boolean matrix.

Name	Notation	$(u_1\|u_2)$ adj $(v_1\|v_2)$ iff	$A(G_1 \circ G_2)$
Conjunction	$G_1 \wedge G_2$	$((u_1, v_1) \in E_1)$ and $((u_2, v_2) \in E_2)$	$A_1 * A_2$
Cartesian Product	$G_1 \times G_2$	$((u_1 = v_1)$ and $((u_2, v_2) \in E_2))$ or $((u_2 = v_2)$ and $((u_1, v_1) \in E_1))$	$(A_1 * I_{N_2}) \oplus (I_{N_1} * A_2)$
Generalized Product	$G_1 \star G_2$	$((u_1 = v_1)$ and $((u_2, v_2) \in E_2))$ or $((v_2 = f_{(u_1 \mapsto v_1)}(u_2))$ and $((u_1, v_1) \in E_1^{dir}))$	
Composition	$G_1[G_2]$	$((u_1, v_1) \in E_1)$ or $((u_1 = v_1)$ and $((u_2, v_2) \in E_2)$	$(A_1 * J_{N_2}) \oplus (I_{N_1} * A_2)$
Symmetric difference	$G_1 \diamond G_2$	$((u_1, v_1) \in E_1)$ xor $((u_2, v_2) \in E_2)$	$(A_1 * J_{N_2}) \oplus (J_{N_1} * A_2)$
Disjunction	$G_1 \vee G_2$	$((u_1, v_1) \in E_1)$ or $((u_2, v_2) \in E_2)$	$(A_1 * J_{N_2}) \oplus (J_{N_1} * A_2) \oplus (A_1 * A_2)$
Rejection	$G_1 \setminus G_2$	$((u_1, v_1) \notin E_1)$ and $((u_2, v_2) \notin E_2)$	$\overline{A_1} * \overline{A_2}$

Table 3.10: Boolean operations on Graphs

Operation	Degree of point $(i\|j)$	L
$G_1 \wedge G_2$	$d_1 d_2$	$2L_1L_2$
$G_1 \times G_2$	$d_1 + d_2$	$L_1N_2 + L_2N_1$
$G_1 \star G_2$	$d_1 + d_2$	$L_1N_2 + L_2N_1$
$G_1[G_2]$	$d_1N_2 + d_2$	$L_1N_2^2 + L_2N_1$
$G_1 \diamond G_2$	$d_1N_2 + d_2N_1 - 2d_1d_2$	$L_1N_2^2 + L_2N_1^2 - 4L_1L_2$
$G_1 \vee G_2$	$d_1N_2 + d_2N_1 - d_1d_2$	$L_1N_2^2 + L_2N_1^2 - 2L_1L_2$
$G_1 \setminus G_2$	$(N_1 - d_1 - 1)(N_2 - d_2 - 1)$	$\binom{N}{2} - L_1N_2^2 - L_2N_1^2 + 2L_1L_2$

Table 3.11: Connection costs

In order to guarantee that boolean operations result in a connected graph, the graphs G_1 and G_2 must have certain properties [Hara 66]:

- The conjunction is connected if and only if G_1 or G_2 contain an odd cycle.
- The cartesian product is connected if and only if G_1 and G_2 are both connected.
- The generalized product is connected if and only if G_1 and G_2 are both connected.
- The composition is connected if and only if G_1 is connected.
- The symmetric difference is connected if and only if G_1 or G_2 is connected.
- The disjunction is connected if and only if G_1 or G_2 is connected.
- The rejection is connected if and only if G_1 or G_2 are disconnected.

The number of nodes in a graph $G = G_1 \circ G_2$ is always $N = N_1N_2$. The node degree and the total number of edges can be calculated as shown in table 3.11.[40] $G_1 \circ G_2$ is regular, only if G_1 and G_2 are regular, but I am not sure whether the same relation is true for symmetry.

The composistion and the conjunction have the highest **connection costs**, the rejection has the lowest[41].

The $G_1 \star G_2$ graph seems to be an interesting candidate for an interconnection network. It provides on the one hand a small degree ($d = d_1 + d_2$), and on the other hand it is possible to

[40] d_1 denotes the node degree of node $i \in G_1$, d_2 denotes the node degree of node $j \in G_2$. L_1 and L_2 denote the total number of links in G_1 and G_2 respectively.

[41] No wonder, since it is usually disconnec ted.

achieve low diameter by choosing an appropriate mapping function. The only disadvantage is that there are no **routing algorithms** for $G_1 \circ G_2$ graphs in the literature.

All structures are not **extensible**.

3.9 Random Graphs

A Random graph is the result of a well defined probabilistic experiment. The probability space is a set of graphs and the probability for the occurence of a certain graph is determined by the conditions of the experiment [Tinh 80].

In the simplest model the probability space contains all possible graphs with N nodes and L links and each graph is assigned the same probability. Such a *random graph* can be constructed by starting with a set of N isolated nodes and succesively adding edges at random. If the resulting graph does not fulfill the desired properties (e. g. the graph is disconnected) it will be discarded and the construction process starts again.

This method has been used by Prior et al. [Prio 90] with a few restrictions in order to achieve certain properties of the resulting graph. First they restrict the construction to a given degree d. If a node is "full" (i. e. has already d adjacent nodes) then no further links may be assigned to this node.

In order to prevent the production of disconnected graphs, the authors suggest to start with a Hamiltonian Cycle of length N instead of N isolated nodes, this also decreases the size of probability space. The resulting structure is called a *Hamiltonian graph* and can be seen as a degenerated Extended Chordal Ring (see definition 36 on page 37) with period $p = N$.

Once some good graphs have been found it is possible to improve their properties (decrease diameter or increase fault tolerance, for instance). Two methods for improvement are suggested:

- Genetic algorithms: A population of graphs is produced by the mechanisms of crossing parents with preferential selection and simple mutations [Norm 88].
- 2-opt: This method tries to improve a graph with respect to certain criteria by exchanging pairs of links. If no further improvement can be achieved the graph is called 2-opt.

A lot of other models of Random graphs can be found in [Tinh 80]. Bollobas [Boll 85] has investigated the problem of determing the properties of random graphs and defining a probability space so that a graph chosen of this space will be very likely to possess a desired property. The results achieved so far are of minor practical use for multicomputer systems. Some examples with good results on Random graphs can be found in [Norm 89]. Those results have shown that randomly constructed graphs can outperform "classical" structures such as hypercubes, trees and so on. They show in general better values for diameter and average distance (see for detail the figures in [Prio 90]) but their practical use is doubtly. On the one hand the lack of explicit rules for connecting nodes makes it difficult to construct, build or analyse such topologies, and on the other hand the asymmetric structure poses a severe problem for routing, mapping and for the design of efficient algorithms.

Chapter 4

Comparison of Topologies

"si parva licet componere magnis"
Vergil

We now want to compare the strengths and weaknesses of the topologies by determing the rates of increase of two key factors as N increases. These factors are connection costs (measured again in the total number of links) and communication delays (measured in diameter). A similar classification sc heme was used in [Witt 81], but only a few topologies were considered (ring, tree, star, hypercube, mesh, Cube-Connected-Cycles and completely connected networks).

Tables 4.1 and 4.2 summarize the dependencies of L and k upon the network size N. If the size of a network N is determined by more than one parameter, there are several possibilities to increase N, which may result in different dependencies. In those cases we will show the change in the measures, when only one parameter is increased in order to increase N and the others remain constant.

It seems less useful to show the exactly quantified change in the measures. On the one hand we already gave closed forms for calculating the measures in the previous section whenever it was possible, on the other hand these tables are devoted to show the most outstanding features at a first glance. Only the order of the dependencies upon N will be shown.

The total number of links increases only linear with N in the optimum, and with N^2 in the worst case. It is most desirable, to have the diameter independent from N (denoted by "fix" in the tables), in the worst case diameter grows at an order of N.[1]

In addition, tables 4.1 and 4.2 shall serve as a brief survey and reference list. In the first column the abbreviation of the topology is shown (see Appendix B.2 for explanations). In the second column a page reference is given, where the reader can find a discussion of the topology in this work, and in the third column a reference to the literature is given.

[1] For some topologies it is not possible to predict the exact values of diameter, so it was necessary to estimate the order of magnitude of the dependence. We will use the symbol ">" to indicate that the real order of magnitude is larger than the given one, and a "<" to indicate that the real order of magnitude is smaller.

	reference to			order of dependence for	
	page	literature	N	L	K
$BG(b,n)$	22	[Rals 82]	b^n		
if b varies				$N\sqrt{N}$	fix
if n varies				N	$\log N$
$BQ(n,s)$	39	[Yous 90]	$(n+1)s^n$		
if n varies				$N \log N$	$\log N$
if s varies				$N\sqrt{N}$	$>$ fix
$BSG(n)$	32	[Aker 84]	$n!$	$> N$	$>$ fix
$C(N)$	19		N	N	N
$C1(d,m)$	51	[Memm 82]	$m(d/2)^{m-1}$		
if d varies				$N\sqrt{N}$	fix
if m varies				N	$\log N$
$CCC(n)$	27	[Prep 81]	$n2^n$	N	$logN$
$CR(w)$	36	[Arde 81]	N even	N	$\sqrt{N}$
$\mathcal{DT}^m(B,h,p,K)$	48	[Ramk 88]	$p\frac{(B+1)B^h-2}{B-1}$		
if B varies				$N\sqrt{N}$	fix
if h varies				N	$\log N$
$DCR(w_1,w_2)$	38	[Yang 88]	N even	N	$\sqrt{N}$
$ECR(W)$	37	[Doty 82]	N	N	$< \sqrt{N}$
$FQ(n)$	40	[El A 91]	2^n	$\log_2 N$	$\log_2(N/2)$
$FT(d,h)$	44	[Frie 66]	$\frac{2(B+1)(B^h-1)}{B-1}$		
if d varies				$N\sqrt{N}$	fix
if h varies				N	$\log N$
$GCR(W)$	42	[Berm 86]	N	N	$< \sqrt{N}$
$GHC(\vec{m})$	30	[Bhuy 84]	$\prod_{i=1}^{n} m_i$	N	$\sqrt{N}$
$GQ(R,n)$	28	[Huan 88]	$R2^{n-1}$		
if R varies				N	N
if n varies				$N \log N$	$\log N$
$HC(W,n)$	29	[Hawk 85]	W^n		
if W varies				$N\sqrt{N}$	fix
if n varies				$N \log N$	$\log N$
$HT(h)$	35	[Good 81]	2^h	N	$\log N$
$K(N)$	20		N	N^2	fix
$KG(b,n)$	23	[Kaut 68]	$b(b-1)^{n-1}$		
if b varies				$N\sqrt{N}$	fix
if n varies				N	$\log N$
$KT(d,h)$	44	[Korn 67]	$\frac{2(2B^{h+1}-(B+1))}{B-1}$		
if d varies				$N\sqrt{N}$	fix
if h varies				N	$\log N$

Table 4.1: Connection costs and Communicaton delays (1)

	reference to			order of dependence for	
	page	literature	N	L	K
$L(N)$	19		N	N	N
$M(W,n)$	29		W^n		
if W varies				N	$N^{1/n}$
if n varies				$N \log N$	$\log N$
$MDMW(N)$	38	[Beiv 87b]	$b(2b-h)$	N	$\sqrt{N/2}$
$MG(n)$	23	[Lela 82]	2^n	N	$< \log N$
$MQ(n)$	27	[Efe 88]	2^n	$N \log N$	$\frac{\log N}{2}$
$MTS(m,B,h))$	49	[Arde 78]	$m(1+\frac{B^{h-1}}{B-1})$		
if m varies				N	not known
if B varies				$N\sqrt{N}$	fix
if h varies				$N \log N$	not known
$NBG(d,l,m_1,m_2)$	45	[Stor 70]	$f(d,l,m_1,m_2)$		
if d varies				$N\sqrt{N}$	fix
if l varies				N	$\log N$
$NHG(m_1,m_2,d,h)$	45	[Stor 70]	$f(d,k,m_1,m_2)$		
if d varies				$N\sqrt{N}$	fix
if l varies				N	$\log N$
$OG(d)$	22	[Aker 65]	$\binom{2d-1}{d}$	$> N$	$< \log N$
$OT(h)$	46	[Nath 83]	$2^{2h}+2^{h+1}(2^h-1)$	N	$\log N$
$PG(n)$	33	[Aker 84]	$n!$	$> N$	not known
$Q(n)$	25	[Peas 77]	2^n	$N \log N$	$\log N$
$S(N)$	21		N	N	fix
$SG(n)$	32	[Aker 87]	$n!$	$> N$	$< \sqrt{N}$
$SOS^n(n)$	34	[Aker 84]	$n!$	N	$< \sqrt{N}$
$SOS^{n-1}(n)$	34	[Aker 84]	$n!$	N	$< \sqrt{N}$
$ST(h)$	36	[Li 86]	2^h-1	N	$\log N$
$T^{full}(B,H)$	20		$\frac{B^h-1}{B-1}$		
if B varies				$N\sqrt{N}$	fix
if h varies				N	$\log N$
$Tor(m,r,p_{up},p_{down})$	50	[Cont 83]	mr		
if m varies				N	$< \sqrt{N}$
if r varies				N	$< \sqrt{N}$
$TQ(n)$	26	[Esfa 88]	2^n	$N \log N$	$\log N$

Table 4.2: Connection costs and Communicaton delays (2)

We will now rate the topologies according to their

1. regularity, the topology is rated with

   + if it is regular, and with
   – otherwise.

2. symmetry,

   ++ indicates a completely symmetric structure,
   + indicates a structure which is symmetric to a very high extent,
   ◦ indicates medium symmetry,
   – is given for structures with less (or no) symmetry.[2]

3. extensibility,

   ++ is given, if the topology is of extensibility type 1 (i. e. the topology possesses all three desired properties for extensibility),
   + indicates, that the topology possesses two out of three (types 2,3 or 5),
   ◦ indicates that the topology fulfills only one of them (types 4,6, or7).
   – stands for a topology, which possesses none of the three properties (type 8),

4. routing algorithms, with an analogous rating scheme to extensibility

   ++ is given, if there is a type 1 routing algorithm known (i. e. the topology possesses all three desired properties for routing),
   + indicates, that the topology possesses two out of three (types 2,3 or 5),
   ◦ indicates that the topology fulfills only one of them (types 4,6, or7).
   – stands for a topology, which possesses none of the three properties (type 8),

[2] I admit that this classification is subjective.

	regularity	symmetry	extensibility	routing
$BG(b,n)$	$-$	$\circ$	$-/\circ$	$+$
$BQ(n,s)$	$-$	$\circ$	$\circ$	$++$
$BSG(n)$	$+$	$+$	$-$	$\circ$
$C(N)$	$+$	$++$	$++$	$++$
$C1(d,m)$	$+$	$+$	$\circ/-$	$\circ$
$CCC(n)$	$+$	$++$	$+$	$++$
$CR(w)$	$+$	$+$	$\circ$	$+$
$DT^m(B,h,p,K)$	$+$	$\circ$	$-/\circ$	$\circ$
$DCR(w_1,w_2)$	$+$	$+$	$\circ$	$+$
$ECR(W)$	$+$	$+$	$+$	$+$
$FQ(n)$	$+$	$++$	$-$	$++$
$FT(d,h)$	$+$	$\circ$	$\circ/+$	$\circ$
$GCR(W)$	$+/-$	$\circ$	$+$	$+$
$GHC(\vec{m})$	$+$	$\circ$	$\circ$	$\circ$
$GQ(R,n)$	$+$	$+$	$\circ$	$\circ$
$HC(W,n)$	$+$	$+$	$-/\circ$	$+$
$HT(h)$	$-$	$\circ$	$+$	$++$
$K(N)$	$+$	$+$	$+$	$+$
$KG(b,n)$	$-$	$\circ$	$-/\circ$	$+$
$KT(d,h)$	$+$	$\circ$	$\circ/+$	$\circ$
$L(N)$	$-$	$\circ$	$++$	$++$
$M(W,n)$	$-$	$\circ$	$+/\circ$	$+$
$MDMW(N)$	$+$	$+$	$+$	$+$
$MG(n)$	$+$	$\circ$	$\circ$	$\circ$
$MQ(n)$	$+$	$\circ$	$\circ$	$+$
$MTS(m,d,h))$	$+$	$\circ$		
$NBG(m_1,m_2,d,h)$	$+$	$\circ$	$\circ/+$	$\circ$
$NHG(m_1,m_2,d,h)$	$+$	$\circ$	$\circ/+$	$\circ$
$OG(d)$	$+$	$++$	$-$	$+$
$OT(h)$	$-$	$\circ$	$\circ/+$	$\circ$
$PG(n)$	$+$	$+$	$-$	$\circ$
$Q(n)$	$+$	$+$	$-$	$+$
$S(N)$	$-$	$+$	$+$	$++$
$SG(n)$	$+$	$+$	$-$	$+$
$SOS^n(n)$	$+$	$+$	$-$	$+$
$SOS^{n-1}(n)$	$+$	$+$	$-$	$+$
$ST(h)$	$+$	$-$	$\circ$	$\circ$
$T^{full}(B,H)$	$-$	$\circ$	$+/\circ$	$++$
$Tor(m,r,p_{up},p_{down})$	$+$	$\circ$	$\circ$	$\circ$
$TQ(n)$	$+$	$\circ$	$\circ$	$+$

Table 4.3: Regularity, Symmetry, Extensibility and Routing

Chapter 5

The Moore Bound: an (unreachable) Challenge

"ignoramus et ignorabimus"
Du Bois-Reymond

In Section 2.1 we have presented the Moore Bound defining an upper bound for the order of a graph with given degree and diameter. We have also transformed this inequality in order to obtain a lower bound for diameter and it is also possible to establish a lower bound for the degree. So we can pose the following three problem formulations:

1. Find a graph G of maximum order for given degree and diameter.
2. Find a graph G with minimum degree for given N and k.
3. Find a graph G with minimum diameter for given N and d.

Clearly all formulations describe the same problem but from a different point of view. There is only a few literature dealing with the problem of minimizing diameter (e. g. [Ples 81], [Imas 81, Imas 83]) and even less dealing with the last problem[1], but a lot of articles deal with the problem of maximizing the number of nodes in a network with given degree and diameter ([Lela 81],[Memm 82] or [Berm 86] for example). So we will first give a survey of the investigations in this research area (Section 5.1).

In Section 5.2 we will retransform the results of the previous section in order to obtain construction methods for minimum diameter networks for some specific combinations of d and N and will on the other hand try to find some constructions which are defined for (nearly) arbitrarily values of N (incremental extensible topologies). We will also take the average distance into consideration because it is important that a multicomputer system does not only have short communication delays in the worst case but also on average.

In the last section we will discuss the issue of minimizing the degree of a network.

5.1 Maximum Node Number

The problem of finding a graph of maximum order first posed by Elspas [Elsp 64] has been discussed in the literature under the name (d, k) problem. A (d, k) graph is defined as a graph of diameter k whose nodes have degree d at most. The problem is to find a construction method which yields

[1] I have found only one article, with the related problem of minimizing the number of edges in a graph with given degree and diameter [Fure 90])

largest possible values for N. We will denote the number of nodes that can be achieved by using a certain construction method by $N_{method}(d,k)$, where "method" identifies the construction.

In Section 2.1 we have presented the Moore Bound as an upper bound for the number of nodes and we will first investigate for which combinations of (d,k) this bound can be reached.

In the following we will discuss some construction methods[2], where we will distinguish 3 different types of construction:

- special constructions, i. e. N_{method} is only defined for one specific pair (d,k)
- one-parameter constructions, i. e. N_{method} is a function of one parameter (d or k) and the other parameter is either constant or also determined by the first parameter, i. e. only a subset of (d,k)-combinations is possible
- general constructions, i. e. N_{method} is a function of d and k, defined for all possible pairs (d,k)

In the last section we will show the present state of the art and show the progress that has been made in this field.

5.1.1 Moore Graphs

A graph is called a *Moore graph* if it reaches the Moore Bound:

$$N_{Moore} = \begin{cases} 2k+1 & \text{if } d = 2 \\ \frac{d(d-1)^k - 2}{d-2} & \text{if } d > 2 \end{cases}$$

and a graph which comes close to this bound will be called *dense*, i. e. there might still exist better graphs although not yet found.

It was shown by Hoffman and Singleton [Hoff 60] that Moore graphs can only exist for the following (d,k)-combinations (see figure 5.1):

- $N_{opt}(d,1) = d+1$: complete graphs,
- $N_{opt}(2,k) = 2k+1$: rings,
- $N_{opt}(3,2) = 10$: Petersen graph,
- $N_{opt}(7,2) = 50$: Singleton graph,
- and possibly[3] $N_{opt}(57,2) = 3250$.

Though it has been proven that the Moore Bound is unattainable for most combinations of d and k, no better upper bound has been found yet. Only for $k = 2$ Erdös, Fajtlowicz and Hoffman [Erdo 80] have shown that $N(d,2) < d^2$ for $d > 3$ and $d \neq 7$. Graphs which reach this bound will be called *maximum* graphs, i. e. although the Moore Bound is not reached there cannot exist any graph with more nodes. For $d = 4$ and $d = 5$ these maximum graphs have already been found:

- $N(4,2) = 15$ is reached by a graph product $(K_3 \star C_5)$ [Berm 84] or by an Extended Chordal Ring [Fers 91b] with

$$W = \begin{pmatrix} 5 & -5 \\ 7 & -4 \\ 4 & 7 \\ -4 & -7 \\ -7 & 4 \end{pmatrix}$$

- $N(5,2) = 24$ is reached by a graph product $(K_3 \star X_8)$ [Berm 82a].

[2] Since we have already presented most of the construction methods in Section 3, we will not explain the methods again but compare the values for N as a function of d and k.

[3] It was neither possible to construct this graph nor to proof its nonexistence.

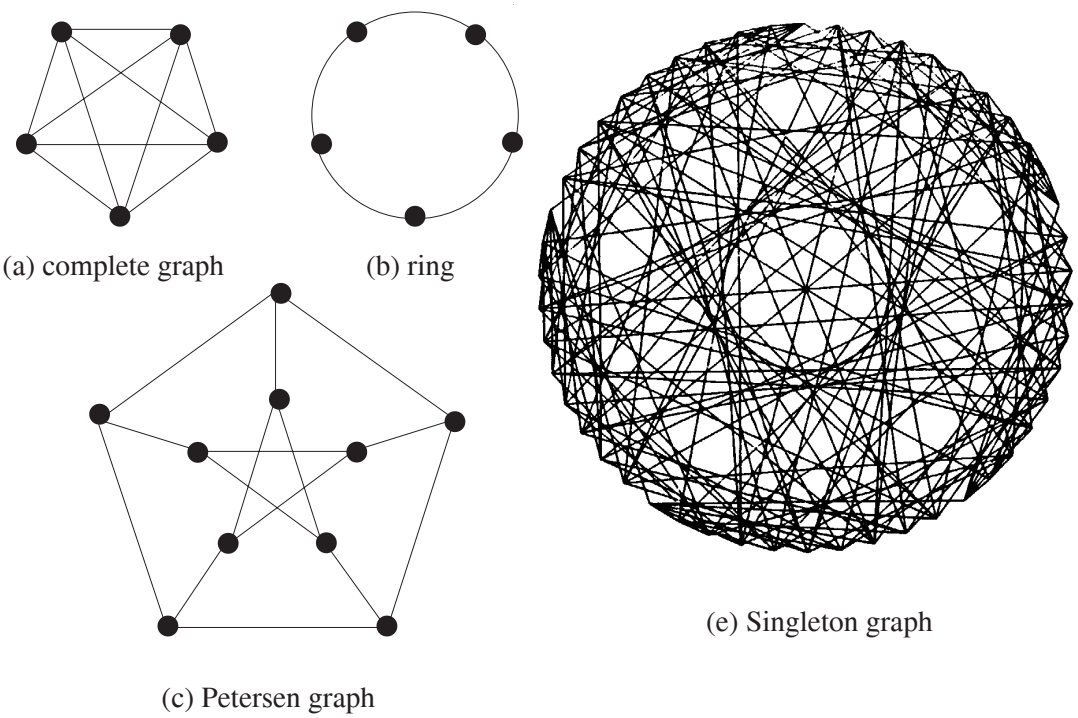


Figure 5.1: Moore graphs

5.1.2 General combinations

Friedman [Frie 66] gave the first construction method for arbitrary combinations of d and k (see page 44):

$$N_{Friedman}(d,k) = \frac{2(d(d-1)^{\lfloor k/2 \rfloor} - d)}{d-2}$$

and Korn [Korn 67] was able to find an improvement for odd diameter (see page 44 for the construction):

$$N_{Korn}(d,k) = \frac{2(2(d-1)^{\lfloor (k+1)/2 \rfloor} - d)}{d-2}$$

It can be shown, that $N_{Korn} > N_{Friedman}$ only if k is odd (see Appendix A.3).

By combining those methods Storwick [Stor 70] was able to create denser graphs. He proposes two different methods namely the NHG and the NBG (see page 45). Unfortunately no rule is given in his article when to use which construction in order to reach maximum values for N. However, it can be shown that $N_{NBG} > N_{NHG}$ only if $k \bmod 4 \equiv 0$. Since the total number of nodes is determined by the number of roots (m_1 and m_2) in each subtree, we have to find for every (d,k)-combination those values for m_1 and m_2 which maximize N_{NHG} if $k \bmod 4 \not\equiv 0$ and N_{NBG} if $k \bmod 4 \equiv 0$. This is a problem of nonlinear discrete optimization.

Maximum N values for $d = 3, 4, \ldots, 10$ and $k = 3, 4, \ldots, 10$ are shown in table 5.1. I have obtained those results by a search procedure.

Comparing Storwick's results to those of Friedman we observe that $N_{Friedman} > N_{Storwick}$ only if $k \bmod 4 \equiv 0$ and that $N_{Storwick} > N_{Korn}$.

k d	3	4	5	6	7	8	9	10
3	12 (2,1)	14 (2,2)	30 (2,1)	44 (7,1)	66 (2,1)	74 (2,2)	138 (2,1)	216 (2,1)
4	20 (3,1)	24 (3,3)	62 (3,1)	114 (6,1)	188 (3,1)	216 (3,3)	566 (3,1)	996 (3,1)
5	30 (4,1)	36 (4,4)	110 (3,1)	232 (5,1)	442 (3,1)	528 (3,3)	1770 (3,1)	3512 (3,1)
6	42 (5,1)	51 (5,5)	177 (4,1)	447 (5,1)	867 (4,1)	1056 (4,4)	4317 (4,1)	9465 (4,1)
7	56 (6,1)	68 (6,6)	266 (4,1)	768 (4,1)	1574 (4,1)	1964 (4,4)	9422 (4,1)	22836 (4,1)
8	72 (7,1)	88 (7,7)	380 (5,1)	1228 (4,1)	2592 (5,1)	3280 (5,5)	18076 (5,1)	47880 (5,1)
9	90 (8,1)	110 (8,8)	522 (5,1)	1904 (3,1)	4098 (5,1)	5270 (5,5)	32706 (5,1)	94416 (5,1)
10	110 (9,1)	135 (9,9)	695 (6,1)	2780 (2,1)	6095 (6,1)	7920 (6,6)	54695 (6,1)	170685 (6,1)

Table 5.1: Dense NBG and NHG

Memmi and Raillard [Memm 82] found further improvements with two construction methods. The first one is called C1 and discussed in Section 3.7.2, the second one is called C2 (only defined for odd diameter) resulting in a graph equivalent to a $\mathcal{D}_1$-tree (see definition 48 on page 48) with single-rooted subtrees of height $h = \lfloor k/2 \rfloor$.

$$N_{C1} = \begin{cases} 3d^2/4 & \text{if } k = 3 \\ \frac{2k+4}{3}(d/2)^{(2k+1)/3} & \text{if } k \bmod 3 \equiv 1 \\ \frac{2k+5}{3}(d/2)^{(2k+2)/3} & \text{if } k \bmod 3 \equiv 2 \end{cases}$$

$$N_{C2} = \frac{d}{d-1}(d(d-1)^{\lfloor k/2 \rfloor} - 2)$$

Comparing both constructions shows that $N_{C1} > N_{C2}$ if $d \geq 4$ and d even $k \geq 7$ and $k \bmod 3 \not\equiv 0$. Furthermore $N_{C2} > N_{NHG}$ if $d \geq 3$ and $k \geq 4$.

The introduction of Extended Chordal Rings [Doty 82] leads to a great progress in the (d, k) problem. Many values could be improved by ECRs and for some combinations of d and k they are still unsurpassed (see table 5.7. The disadvantage of ECRs is that there is no closed form for the number of nodes for given degree or diameter, neither is there a rule for determing the optimum chord matrix. The ECRs in table 5.2 have been found by random search[4], so there might still be even better ECRs. The values in the parenthesis show the corresponding chord matrices.

Delorme [Delo 84, Delo 85] achieved some good results with boolean operations on graphs and in [Carl 85] a generalization of Cube connected Cycles is proposed, which are still among the densest known graphs (compare table 5.6).

An interesting structure proposed by Ramkumar [Ramk 88] are $\mathcal{D}_K$-trees. As we see from the definition of $\mathcal{D}_K$-trees (see page 48) the number of nodes in a $\mathcal{D}_K$-trees depends on the subgraph used for leaf interconnection. This graph consists of p nodes with diameter K and degree $d-1$ so the problem of finding a maximum $\mathcal{D}_K$-tree consists in finding a maximum graph for degree $d-1$ and diameter K, which we will denote by $p(d-1, K)$. The number of nodes also depends on whether single or double rooted subtrees are used. In the case of single rooted subtrees the diameter of a $\mathcal{D}_K$-tree is given by $k = 2h + K$ and therefore the possible values for K for a given diameter are in a range of $1 \leq K \leq k-2$ and if k is even then also K must be an even number (analogously for odd numbers). If double rooted subtrees are used instead, the diameter is given by $k = 2h + K + 1$ and we have $1 \leq K \leq k-3$ and if k is even (odd) K must be odd (even).

I was not able to find analytically those K which maximize the number of nodes for any given conbination of (d, k). So I have applied exhaustive search instead, the results are shown in table 5.3, where I have used the values in table 5.7 for p. The values in parenthesis show the corresponding optimum K. It is easy to see that all those $\mathcal{D}$-trees are made of single rooted subtrees (remember the even-odd relation mentioned above), the double rooted ones have always been smaller.

[4]Some results are taken from [Doty 84], some results have been found by myself.

d k	3	4	6	7
3	12 (6)	36 $\begin{pmatrix} 4 & 20 \\ 10 & 32 \\ 16 & 26 \end{pmatrix}$		122 $\begin{pmatrix} 111 & 25 & 39 & 114 & 8 \\ 97 & 52 & 83 & 11 & 70 \end{pmatrix}$
4	28 $\begin{pmatrix} 6 \\ 10 \\ 18 \\ 14 \end{pmatrix}$	64 $\begin{pmatrix} 7 & 56 \\ 16 & 77 \\ 28 & 68 \end{pmatrix}$		420 $\begin{pmatrix} 321 & 158 & 13 & 382 & 99 \\ 10 & 407 & 38 & 229 & 325 \\ 95 & 191 & 410 & 262 & 210 \end{pmatrix}$
5	56 $\begin{pmatrix} 10 \\ 26 \\ 46 \\ 30 \end{pmatrix}$	128 $\begin{pmatrix} 22 & 106 \\ 56 & 72 \end{pmatrix}$	820 $\begin{pmatrix} 673 & 63 & 279 & 367 \\ 568 & 106 & 187 & 593 \\ 475 & 345 & 453 & 714 \\ 147 & 757 & 536 & 633 \\ 252 & 227 & 284 & 541 \end{pmatrix}$	1550 $\begin{pmatrix} 1013 & 226 & 782 & 1084 & 937 \\ 832 & 673 & 134 & 193 & 692 \\ 1131 & 768 & 842 & 613 & 1331 \\ 718 & 537 & 219 & 858 & 419 \\ 877 & 466 & 1357 & 708 & 775 \end{pmatrix}$

k d	6	7	8	9	10
3	84	180 $\begin{pmatrix} 153 \\ 16 \\ 116 \\ 27 \\ 64 \\ 164 \end{pmatrix}$	280 $\begin{pmatrix} 146 \\ 101 \\ 73 \\ 17 \\ 179 \\ 207 \\ 26 \end{pmatrix}$	462 $\begin{pmatrix} 231 \\ 37 \\ 16 \\ 139 \\ 247 \\ 425 \\ 79 \\ 446 \\ 383 \\ 215 \\ 323 \end{pmatrix}$	708 $\begin{pmatrix} 632 \\ 685 \\ 23 \\ 208 \\ 601 \\ 107 \\ 483 \\ 500 \\ 76 \\ 225 \\ 433 \\ 275 \end{pmatrix}$
4		856 $\begin{pmatrix} 260 & 492 \\ 386 & 444 \\ 67 & 412 \\ 470 & 684 \\ 364 & 596 \\ 789 & 412 \\ 417 & 444 \\ 439 & 172 \end{pmatrix}$	1872 $\begin{pmatrix} 505 & 1795 \\ 249 & 1367 \\ 1563 & 1830 \\ 860 & 642 \\ 77 & 1100 \\ 42 & 479 \\ 1230 & 772 \\ 1623 & 1393 \\ 309 & 1012 \end{pmatrix}$	3708 $\begin{pmatrix} 1938 & 2603 \\ 294 & 56 \\ 3694 & 1105 \\ 3652 & 2028 \\ 1769 & 442 \\ 3266 & 3387 \\ 1680 & 14 \\ 3187 & 3414 \\ 512 & 321 \end{pmatrix}$	7100 $\begin{pmatrix} 2165 & 2057 \\ 2500 & 4600 \\ 1031 & 1675 \\ 6069 & 7086 \\ 725 & 262 \\ 5893 & 4935 \\ 2932 & 4448 \\ 5425 & 5043 \\ 1207 & 4168 \\ 14 & 6375 \end{pmatrix}$
5	60 $\begin{pmatrix} 4 & 37 & 47 \\ 19 & 56 & 23 \\ 41 & 13 & 30 \end{pmatrix}$				

Table 5.2: Dense Extended Chordal Rings

k d	3	4	5	6	7	8	9	10
3	12 (1)	20 (2)	30 (1)	50 (2)	70 (3)	110 (2)	154 (3)	230 (2)
4	20 (1)	50 (2)	100 (3)	190 (4)	350 (5)	646 (4)	1190 (5)	21766 (6)
5	30 (1)	90 (2)	240 (3)	570 (4)	2184 (5)	4386 (6)	9464 (5)	19006 (6)
6	42 (1)	168 (2)	490 (3)	1218 (4)	3724 (5)	19138 (6)	32808 (6)	101158 (6)
7	56 (1)	256 (2)	840 (3)	2560 (4)	7936 (5)	62536 (6)	104448 (7)	390850 (6)
8	72 (1)	450 (2)	1098 (3)	4320 (4)	13950 (5)	94914 (6)	316386 (7)	843696 (8)
9	90 (1)	570 (2)	2000 (3)	8070 (4)	28080 (5)	392230 (6)	703080 (7)	3216286 (6)
10	110 (1)	814 (2)	6435 (3)	13728 (4)	59085 (5)	823966 (6)	2372568 (7)	7565506 (6)

Table 5.3: Dense $\mathcal{D}_K$-trees

k	$p=1$ N	W	$p=2$ N	W	$p=3$ N	W	$p=4$ N	W
2	8	(4)	6	($\pm$3)				
3	12	(6)	14	($\pm$5)	12	(6, $\pm$4)	12	($\pm$3, $\pm$5)
4	16	(8)	20	($\pm$7)	24	(12, $\pm$7)	28	($\pm$6, $\pm$10)
5	20	(10)	34	($\pm$9)	48	(24, $\pm$16)	56	($\pm$10, $\pm$26)
6	24	(12)	44	($\pm$9)	72	(36, $\pm$22)	84	($\pm$10, $\pm$30)
7	28	(14)	64	($\pm$11)	102	(51, $\pm$40)	140	($\pm$17, $\pm$57)
8	32	(16)	70	($\pm$13)	132	(66, $\pm$52)	196	($\pm$10, $\pm$78)
9	36	(18)	102	($\pm$13)	168	(84, $\pm$13)		
10	40	(20)	118	($\pm$15)	216	(108, $\pm$46)		

Table 5.4: Dense Extended Chordal Rings of degree 3

5.1.3 One Parameter Constructions

Most of the one-parameter constructions have been proposed for cubic graphs (graphs of degree 3). Those graphs are of special interest because of their low connection costs.

The Cube Connected Cycles network is a cubic graph with

$$N_{CCC}(3, \lceil \frac{5n-5}{2} \rceil) = n2^n$$

for $n > 3$.

Orthogonal tree networks are restricted to $k \bmod 4 \equiv 0$ and $k \geq 4$ with

$$N_{OTN}(3,k) = 2^{k/2} + 2^{1+k/4}(2^{k/4} - 1)$$

Another construction for cubic graphs of diameter $k = 2$ or $k > 3$ was proposed by Bermond et.al. [Berm 84] (see Section 3.8) with

$$N_{Bermond}(3,k) = \begin{cases} 10 & \text{if } k = 2 \\ 2(2k^2 - 6k + 5) & \text{if } k > 3 \end{cases}$$

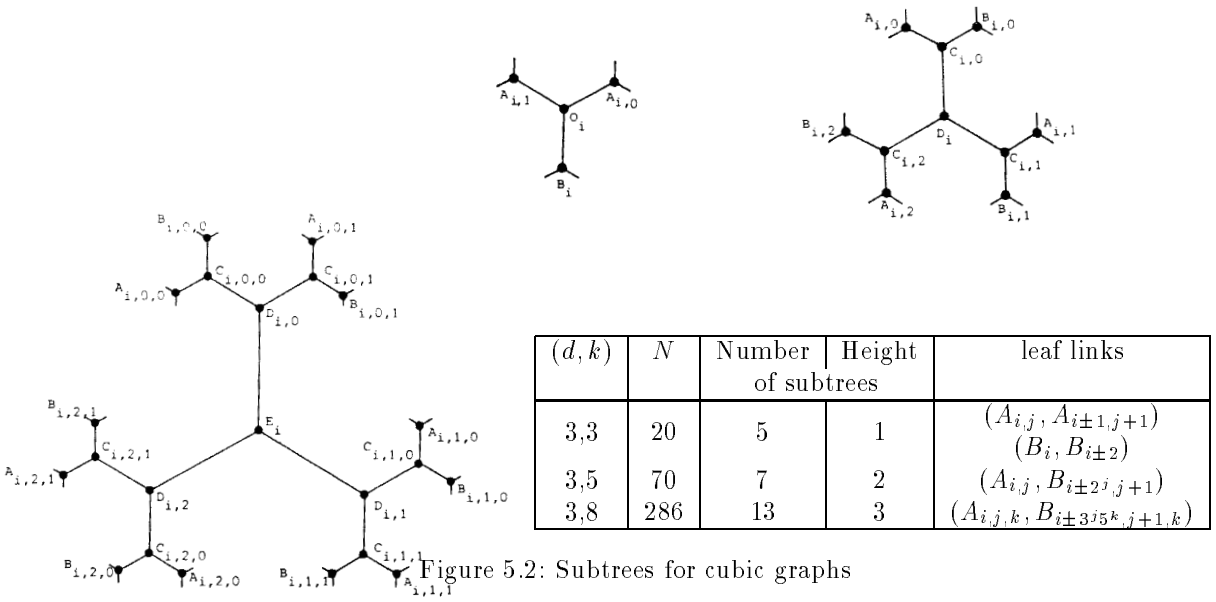
In [Arde 81] Chordal Rings of period $p = 2$ are investigated under the problem of maximizing the number of nodes for given values of diameter and $d = 3$. The maximum number of nodes and the corresponding optimum chord length are given as functions of diameter:

$$N_{Arden}(3,k) = \begin{cases} 6 & \text{if } k = 2 & \text{with } w = 3 \\ 14 & \text{if } k = 3 & \text{with } w = 5 \\ k^2 + 3k - 4 & \text{if } k = 4 \text{ or } k = 6 & \text{with } w = k + 3 \\ k^2 + 3k - 6 & \text{if } k \geq 5 \text{ and odd} & \text{with } w = k + 4 \\ k^2 + 3k - 12 & \text{if } k \geq 8 \text{ and even} & \text{with } w = k + 5 \end{cases}$$

Experimental results have shown that there are even larger Chordal Rings (compare the column $p = 2$ in table 5.4, which shows densest Chordal Rings found by myself), but I have not been able to find the corresponding formulas for calculating those rings.

The extension to higher periods (see table 5.4 will bring a further improvement, but again no closed form for calculating the number of nodes and the corresponding chord lengths reaching this maximum could be found. It is only possible to try out all possibilities in order to find a graph with mimimum diameter.

A further improvement can be achieved by Generalized Chordal Rings where the restriction that all nodes lie on a cycle is weakened. But the number of possible rings grows rapidly with p and k



(d,k)	N	Number of subtrees	Height	leaf links
3,3	20	5	1	$(A_{i,j}, A_{i\pm1,j+1})$ $(B_i, B_{i\pm2})$
3,5	70	7	2	$(A_{i,j}, B_{i\pm2^j,j+1})$
3,8	286	13	3	$(A_{i,j,k}, B_{i\pm3^j5^k,j+1,k})$

Figure 5.2: Subtrees for cubic graphs

k	CR	ECR	CCC	OTN	Bermond	Alegre
2	6	8			10	
3	14	12				20
4	24	28		8	26	
5	34	56	24		50	70
6	50	84			82	
7	64	140			122	
8	76		64	40	170	386

Table 5.5: Dense cubic graphs

and exhaustive search becomes soon computationally infeasible . I also suspect, that the increase in the number of nodes is not significant. For instance, the maximum ECR with period 3 and diameter 6 consists of 70 nodes and the optimum GCR with $p = 3$ and $k = 6$ consists of 78 nodes, both rings are of degree 3.

Alegre [Aleg 86] proposed a construction for cubic graphs with diameters 3,5, and 8. The graphs are obtained from maximum binary trees whose leaves[5] are linked according to certain rules as shown in the table in figure 5.2). The $(3,3)$ graph is made of 5 subtrees of height 1, the $(3,5)$ graph consists of 7 subtrees of height 2, and the graph with diameter 8 is made of 13 subtrees of height 3.

In table 5.5 we will compare the order of all those cubic graphs. CCC and OTN seem to be very poor, but if we look at the asymptotic behaviour of N for increasing k, those constructions are superior because N grows at a power of 2, whereas the number of nodes in CRs and in the construction of Bermond only grows quadratically with k. The construction of Alegre is only defined for 3 values of k, but those 3 graphs are much larger than the other cubic graphs with equal diameter. Extended Chordal Rings also show comparably good results but the lack of an explicit construction rule is a disadvantage since it is necessary to find an optimum ECR by exhaustive (or random) search.

5.1.4 State of the art

Table 5.6 summarizes the progress in the (d,k) problem in the last 20 jears. We can see that there is a great progress in this field but the results are still far away from the Moore Bound. Most of the

[5]The leaves are labelled with the letters A, and B, the first index denotes the number of the subtree, the second index denotes the number of the leaf within a subtree.

(d,k)	1970		1982		1984		1986		Moore
(3,8)	90	[Frie 66]	124	[Arde 78]	280	[Doty 84]	286	[Aleg 86]	766
(4,5)	62	[Stor 70]	80	[Memm 82]	364	[Berm 82b]	364		485
(5,9)	1,770	[Stor 70]	2,130	[Memm 82]	11,340	[Berm 82b]	15,360	[Carl 85]	436,906

Table 5.6: Progress in maximizing the number of nodes

d k	3	4	5	6	7	8	9	10
2	10	15	24	32	50	57	74	91
3	20	41	70	105	136	203	585	650
4	38	95	184	355	506	915	1254	1820
5	70	364	532	1088	2460	4108	6890	12144
6	130	734	2742	7832	10554	39258	74954	132932
7	184	1081	4380	14878	41024	104808	217622	490052
8	320	2943	12246	53368	150000	481179	1320000	3000000
9	540	7439	41684	210000	911088	2400000	4965098	9000000
10	938	15657	132000	900000	4773696	7738848	19845936	47059200

Table 5.7: Current Densest graphs

densest graphs known today are the results of boolean operations on graphs. But the practical use of those graphs for multicomputer networks is doubtly. On the one hand there is minor research done on investigating properties other than the diameter, so we do know a few about the fault tolerance and other important measures. On the other hand no routing strategies for those graphs have been proposed, and the use of routing tables (see Section 2.3) is inpracticable for networks with more then 1000 nodes. So it might be more important to develop efficient routing schemes (see Chapter 6) for existing topologies than to invent even larger and more complex graphs.

In table 5.7 a table of largest known (d,k) graphs is given taken from [Berm 91a]. Unfortunately, I was not able to receive an updated version of this table.

5.2 Minimum Diameter

In the previous section we have shown constructions for finding maximum (densest) graphs. Most of these graphs are extensible only in large steps, i. e. there are huge gaps in the range of N if we want to use those constructions for determing a network with minimum diameter. In this section we want to close these gaps by some other constructions which are not as good as the ones from the previous section, but have the advantage that they are applicable for (nearly) arbitrary values of N.

As mentionend at the beginning of this section we will also look at the average distance in the networks. A network of order N and degree d with minimum diameter k does not automatically have minimum average distance if $N < N_{Moore}(d,k)$, i. e. there might be several graphs which have the same optimum diameter but different values for average distance. There are also some rare examples where a network with a larger value for diameter will have a smaller average distance ([Cont 83]). However, we will in general neglect those exceptional cases and try to find networks with minimum diameter and will use the average distance as a second criterion to choose among networks with the same diameter.

Toueg and Steiglitz [Toue 79] have found some good results by using a local search algorithm. This algorithm starts with an arbitrary graph which is then successively improved. So it is rather a tuning method than a construction. The improvement is done by performing a twist operation on two disjoint edges (called *X-change*) and the criteria for improvement is either diameter or average

N	degree 3		degree 4	
	k	μ	k	μ
8	2	1.571	2	1.429
16	3	2.200	3	1.750
24	4	2.565	3	2.130
32	5	2.980	3	2.355
40	5	3.183	4	2.504
48	5	3.411	4	2.644

Table 5.8: Graphs with minimum diameter and average distance found by local search

N	$p=1$		$p=2$		$p=3$		$p=4$	
	k	μ	k	μ	k	μ	k	μ
12	3	2.09	3	2.00	3	2.03	3	2.00
	(6)		(5 7)		(6 4 8)		(5 7 10 3)	
24	6	3.69	5	2.78	4	3.57	5	2.67
	(12)		(7 17)		(12 7 17)		(6 10 18 14)	
36	9	5.11	6	3.37	5	3.10	5	3.07
	(18)		(7 29)		(18 7 29)		(6 14 30 22)	
48	12	6.62	6	3.83	5	3.54	5	3.44
	(24)		(19 29)		(24 16 32)		(6 18 42 30)	
60	15	8.12	7	4.27	6	3.82	6	3.77
	(30)		(17 43)		(30 22 38)		(10 26 50 34)	
72	18	9.62	8	4.70	6	4.11	6	3.99
	(36)		(11 61)		(36 10 62)		(18 30 54 42)	
84	21	11.12	9	5.08	7	4.37	6	4.22
	(42)		(13 71)		(42 25 59)		(10 30 74 54)	
96	24	12.62	9	5.41	7	4.62	7	4.42
	(48)		(21 75)		(48 19 77)		(10 34 86 62)	
108	27	14.12	9	5.70	8	4.90	7	4.60
	(54)		(41 67)		(54 43 65)		(10 30 98 78)	
120	30	15.62	10	6.02	8	5.12	7	4.74
	(60)		(53 67)		(60 13 107)		(10 46 98 78)	
132	33	17.12	10	6.29	8	5.32	7	4.91
	(66)		(29 103)		(66 52 80)		(17 115 53 79)	
144	36	18.62	11	6.57	9	5.53	8	5.06
	(72)		(43 101)		(72 13 131)		(26 42 118 102)	
156	39	20.12	11	6.86	9	5.73	8	5.17
	(78)		(43 113)		(78 22 132)		(17 139 65 91)	
168	42	21.62	12	7.16	9	5.93	8	5.39
	(84)		(17 151)		(84 13 155)		(13 155 37 131)	
180	45	23.12	12	7.36	10	6.11	8	5.47
	(90)		(49 131)		(90 52 128)		(9 171 111 69)	
192	48	24.62	12	7.57	10	6.29	8	5.70
	(96)		(35 157)		(96 55 137)		(14 70 178 122)	

Table 5.9: Extended Chordal Rings with minimum k and μ for degree 3

N	$p=1$		$p=2$		$p=3$		$p=4$	
	k	μ	k	μ	k	μ	k	μ
12	3	1.73	3	1.73	2	1.64	2	1.64
	$\begin{pmatrix} 4 \\ 8 \end{pmatrix}$		$\begin{pmatrix} 4 & 2 \\ 8 & 10 \end{pmatrix}$		$\begin{pmatrix} 3 & 4 & 5 \\ 9 & 7 & 8 \end{pmatrix}$		$\begin{pmatrix} 4 & 10 & 7 & 7 \\ 8 & 5 & 5 & 2 \end{pmatrix}$	
24	4	2.39	3	2.13	3	2.13	3	2.13
	$\begin{pmatrix} 10 \\ 14 \end{pmatrix}$		$\begin{pmatrix} 4 & 10 \\ 20 & 14 \end{pmatrix}$		$\begin{pmatrix} 4 & 7 & 17 \\ 14 & 20 & 10 \end{pmatrix}$		$\begin{pmatrix} 4 & 10 & 4 & 14 \\ 20 & 14 & 20 & 10 \end{pmatrix}$	
36	4	2.86	4	2.46	3	2.43	3	2.43
	$\begin{pmatrix} 8 \\ 28 \end{pmatrix}$		$\begin{pmatrix} 8 & 16 \\ 28 & 20 \end{pmatrix}$		$\begin{pmatrix} 4 & 10 & 16 \\ 20 & 32 & 26 \end{pmatrix}$		$\begin{pmatrix} 8 & 10 & 16 & 26 \\ 28 & 30 & 206 & \end{pmatrix}$	
48	5	3.30	4	2.77	4	2.66	4	2.67
	$\begin{pmatrix} 20 \\ 24 \end{pmatrix}$		$\begin{pmatrix} 6 & 20 \\ 42 & 28 \end{pmatrix}$		$\begin{pmatrix} 6 & 22 & 11 \\ 42 & 37 & 26 \end{pmatrix}$		$\begin{pmatrix} 8 & 34 & 27 & 39 \\ 40 & 21 & 9 & 14 \end{pmatrix}$	
60	6	3.69	4	2.90	4	2.88	4	2.83
	$\begin{pmatrix} 8 \\ 52 \end{pmatrix}$		$\begin{pmatrix} 24 & 16 \\ 36 & 44 \end{pmatrix}$		$\begin{pmatrix} 6 & 22 & 11 \\ 54 & 49 & 38 \end{pmatrix}$		$\begin{pmatrix} 8 & 26 & 10 & 22 \\ 52 & 28 & 50 & 34 \end{pmatrix}$	
72	6	4.05	4	3.07	4	3.02	4	2.99
	$\begin{pmatrix} 11 \\ 61 \end{pmatrix}$		$\begin{pmatrix} 6 & 28 \\ 66 & 54 \end{pmatrix}$		$\begin{pmatrix} 10 & 26 & 44 \\ 23 & 62 & 49 \end{pmatrix}$		$\begin{pmatrix} 8 & 26 & 20 & 34 \\ 54 & 38 & 52 & 46 \end{pmatrix}$	
84	7	4.36	5	3.23	4	3.13	4	3.13
	$\begin{pmatrix} 32 \\ 52 \end{pmatrix}$		$\begin{pmatrix} 26 & 38 \\ 58 & 46 \end{pmatrix}$		$\begin{pmatrix} 7 & 16 & 26 \\ 56 & 77 & 68 \end{pmatrix}$		$\begin{pmatrix} 8 & 26 & 32 & 14 \\ 76 & 70 & 52 & 58 \end{pmatrix}$	
96	7	4.64	5	3.39	5	3.26	5	3.26
	$\begin{pmatrix} 42 \\ 54 \end{pmatrix}$		$\begin{pmatrix} 8 & 28 \\ 88 & 68 \end{pmatrix}$		$\begin{pmatrix} 7 & 16 & 34 \\ 62 & 89 & 80 \end{pmatrix}$		$\begin{pmatrix} 37 & 6 & 26 & 15 \\ 70 & 59 & 81 & 90 \end{pmatrix}$	
108	7	4.91	5	3.50	5	3.38	5	3.36
	$\begin{pmatrix} 36 \\ 72 \end{pmatrix}$		$\begin{pmatrix} 12 & 20 \\ 96 & 88 \end{pmatrix}$		$\begin{pmatrix} 12 & 39 & 45 \\ 96 & 69 & 63 \end{pmatrix}$		$\begin{pmatrix} 9 & 34 & 86 & 27 \\ 22 & 99 & 81 & 74 \end{pmatrix}$	
120	8	5.18	5	3.63	5	3.47	5	3.45
	$\begin{pmatrix} 14 \\ 106 \end{pmatrix}$		$\begin{pmatrix} 14 & 22 \\ 106 & 98 \end{pmatrix}$		$\begin{pmatrix} 12 & 45 & 51 \\ 108 & 75 & 69 \end{pmatrix}$		$\begin{pmatrix} 9 & 82 & 106 & 71 \\ 14 & 111 & 49 & 38 \end{pmatrix}$	
132	9	5.44	5	3.75	5	3.56	5	3.53
	$\begin{pmatrix} 58 \\ 74 \end{pmatrix}$		$\begin{pmatrix} 16 & 26 \\ 116 & 106 \end{pmatrix}$		$\begin{pmatrix} 12 & 51 & 57 \\ 120 & 81 & 75 \end{pmatrix}$		$\begin{pmatrix} 9 & 94 & 110 & 99 \\ 22 & 123 & 33 & 38 \end{pmatrix}$	
144	9	5.69	6	3.82	5	3.64	5	3.62
	$\begin{pmatrix} 22 \\ 122 \end{pmatrix}$		$\begin{pmatrix} 16 & 60 \\ 128 & 84 \end{pmatrix}$		$\begin{pmatrix} 12 & 57 & 63 \\ 122 & 87 & 81 \end{pmatrix}$		$\begin{pmatrix} 17 & 26 & 94 & 107 \\ 50 & 127 & 37 & 118 \end{pmatrix}$	
156	9	5.90	6	3.83	5	3.72	5	3.69
	$\begin{pmatrix} 46 \\ 110 \end{pmatrix}$		$\begin{pmatrix} 36 & 56 \\ 120 & 100 \end{pmatrix}$		$\begin{pmatrix} 12 & 30 & 54 \\ 144 & 126 & 102 \end{pmatrix}$		$\begin{pmatrix} 9 & 26 & 42 & 31 \\ 114 & 147 & 125 & 130 \end{pmatrix}$	
168	9	6.13	6	4.01	5	3.82	5	3.76
	$\begin{pmatrix} 16 \\ 162 \end{pmatrix}$		$\begin{pmatrix} 22 & 30 \\ 146 & 138 \end{pmatrix}$		$\begin{pmatrix} 9 & 36 & 117 \\ 159 & 132 & 51 \end{pmatrix}$		$\begin{pmatrix} 77 & 42 & 158 & 71 \\ 10 & 91 & 97 & 126 \end{pmatrix}$	
180	10	6.35	6	4.11	6	3.84	5	3.82
	$\begin{pmatrix} 32 \\ 148 \end{pmatrix}$		$\begin{pmatrix} 14 & 80 \\ 166 & 100 \end{pmatrix}$		$\begin{pmatrix} 12 & 30 & 66 \\ 168 & 150 & 114 \end{pmatrix}$		$\begin{pmatrix} 9 & 46 & 82 & 159 \\ 98 & 171 & 21 & 134 \end{pmatrix}$	
192	10	6.56	6	4.19	6	3.91	5	3.89
	$\begin{pmatrix} 88 \\ 104 \end{pmatrix}$		$\begin{pmatrix} 22 & 36 \\ 170 & 156 \end{pmatrix}$		$\begin{pmatrix} 24 & 54 & 66 \\ 168 & 138 & 126 \end{pmatrix}$		$\begin{pmatrix} 9 & 22 & 118 & 111 \\ 74 & 183 & 81 & 17 \end{pmatrix}$	

Table 5.10: Extended Chordal Rings with minimum k and μ for degree 4

N	24	36	48	60	72
$d=3$	4 (12,1,1,1)	5 (18,1,1,1)		5 (15,2,4,4)	6 (36,1,17,17)
$d=4$	4 (12,2,5,5)	3 (9,4,4,7)	4 (12,4,5,5)	4 (15,4,4,4)	5 (18,4,5,11)
N	84	96	108	120	132
$d=3$	7 (21,2,8,8)	7 (24,2,7,7)		7 (15,4,4,4)	8 (11,6,3,4)
$d=4$	5 (14,6,3,5)		5 (18,6,5,11)	5 (30,4,7,13)	
N	144	156	168	180	
$d=3$	8 (18,4,7,13)	8 (39,2,14,14)	8 (14,6,3,5)	9 (18,5,7,13)	
$d=4$		5 (26,6,7,15)	6 (56,3,9,25)	6 (90,2,70,53)	

Table 5.11: Torus networks for $d=3$ and $d=4$

distance[6]. The disadvantages of this method are that the resulting structure will be asymmetric, that no efficient routing algorithm can be given, and that it becomes computationally infeasible for larger values of N ($N > 150$). The results for degree 3 and 4 taken from [Toue 79] are given in table 5.8.

A better method for tuning is offered by Generalized or Extended Chordal Rings. It is possible to decrease diameter and average distance for a given number of nodes by increasing the period of the ring[7] and choosing appropriate chord lengths, so one can make a tradeoff between symmetry and diameter. The problem is to calculate the optimum chord lengths, because the number of possible GCRs grows rapidly with N, d and p.

For degree 3 and 4 and small values of N (< 200) and p (< 5) it is computationally feasible to evaluate all possible Extended Chordal Rings. I have done this for values of N in steps of 12, the results are shown in tables 5.9 and 5.10 [8]. I have chosen the increment of 12, because it guarantees, that N will be a multiple of all periods $p < 5$.

Conta [Cont 83] has proposed a method for graphs of degree 3 and 4 (see definition 50 on page 50). The maximum number of nodes for given diameter is determined by 4 parameters (m,r,p_{up},p_{down}) and the only possibility for determing the optimum values for those parameters is exhaustive search, i. e. trying out all possible combinations. Table 5.11 lists the diameter for some torus networks, with the values in parenthesis showing the 4 parameters, those results are taken from [Cont 83].

5.3 Minimum degree

A lower bound for the degree is given by the following relation:

$$2 - 2N = d_{min}(d_{min} - 1) - N$$

which can be evaluated for a network of given order N and diameter k.

[6] If μ is used as the improvement criteria the algorithm will be able to find those exceptional cases mentioned above where a network with smaller μ will have a larger value for diameter. But among the graphs investigated in [Toue 79] no such abnormality has occured.

[7] with the restriction that the period must be a divisor of N (see the definition of $GCRs$ on page 42.

[8] The corresponding chord matrices are W^T, the transponed of W (i. e. element w_{ij} is the chordlength for the i-th link in period j).

The problem of finding a network with minimum degree is of less practical use when configuring an existing system because the degree will be limited by hardware restrictions. For hardware designers it might be interesting to know how many links per PE are necessary in order to connect a given number of processing elements without exceeding a given bound on diameter. However, there is no literature dealing with this problem. For some specific values of N and k it is possible to use the results of the previous section. For example, the Extended Chordal Ring on 15 nodes, which is maximum, is also a graph with a minimum degree for $N = 15$ and $k = 2$.

A related problem consists in minimizing the total number of links (i.e. the connection costs) in a network of a given order with the restrictions that neither diameter nor degree must exceed a certain value. The minimization of L will lead to irregular structures and to lower fault tolerance. So I do not think, that it is advantageous to minimize L in a multi-computer network topology.

Chapter 6

Routing in Static Networks

"panta rhei"
Heraklit

When executing a parallel program on a multi-computer system, the processing elements will have to exchange information, a process which we call *routing*. According to the number of communication partners we distinguish the following types of routing [Vali 81]:

- point-to-point routing, i. e. one node wants to send a message to another node
- broadcasting, i. e. one node (called *originator* distributes a message to all others)
- gossiping, i. e. each sends a message to all others by simultaneously receiving the messages from all others

When analysing the routing requirements of a specific algorithm, it is also important to know the frequency at which routing requests occur. If, for instance, two nodes want to broadcast a message at the same time, it is necessary to know in advance about possible congestions (on links or nodes), and to implement mechanisms to avoide these congestions (e. g. taking an alternative path). In our discussion we will neglect the frequency of routing requests, and focus on the three basic communication problems mentioned above.

Within each problem class we can further distinguish according to the assumptions made upon what a PE can do concurrently:

- A PE can either send one message to, or receive a message from one of its direct neighbours[1] (half duplex mode).
- A PE can send (receive) a message to (from) one of its direct neighbours (full duplex, single link availability [Bert 91]).
- A PE can send one and the same message to all of its direct neighbours and concurrently receive a message from all of its direct neighbours.
- A PE can send distinct messages to all of its direct neighbours and receive a message from all of its direct neighbours (multiple link availability [Bert 91]).

We will in the following try to develop general communication schemes for all those routing problems. A communication scheme is defined as a pair (G, R) where G is the graph model of the network and R is a routing function defining the path from node i to node j for all nodes

[1] A node i is called a direct neighbour to node j if and only if there is a link connecting both i with j.

in the network [Upfa 84]. A general communication scheme specifies the desired properties, that a graph must possess in order to allow routing in a minimum amount of time, and a routing strategy, describing the routing requirements and outlining the basic principles for solving the routing problem. In Section 6.1 we will start with determing lower bounds on the time necessary for routing[2] under the assumptions of single and multiple link availability. Afterwards we will define general (topology independent) routing strategies and discuss the properties that a graph must possess in order to enable routing in minimum time (general communication schemes). In section 6.2 we will apply those general routing techniques to Extended Chordal Rings and develop routing algorithms for broadcasting and gossiping in ECRs (communication schems for $G = ECR$).

6.1 Topology independent routing

The basic process in all communication problems is the transmission of a message from one node to one of its direct neighbours. The amount of time T required for this basic process depends on a latency necessary for establishing the connection between node i and node j (*start up time* β_{ij}), on the bandwidth of the link ($\frac{1}{\tau_{ij}}$) and on the size of the message (M). We will assume that all links have the same bandwith ($\tau_{ij} = \tau$) and that the start up times $\beta_{ij} = \beta$ are constant for every pair of nodes obtaining $T = \beta + M\tau$ for the amount of time necessary to transmit a message of size M from any node i to its neighbour j.

If all messages are of the same size, T becomes constant and we can simplify the time analysis. The transmission of a single message from any processing element P_i to one of its neighbours P_j takes exactly one unit of time then. If the links are bidirectional P_j can - while receiving from P_i - concurrently send one message to P_i. We will refer to this exchange of information as a *call*. A *step* consists of one or more calls that can be done concurrently and we are interested in minimizing the number of steps necessary for solving a communication problem. We will denote the minimum number of steps by $\varrho_{i:j}$, where i and j specifies the number of communication partners. $1:1$ stands for point-to-point routing, $1:N$ denotes broadcasting, and $N:N$ denotes gossiping.

6.1.1 Point-to-point-routing

Point-to-point communication from node i to node j can be split into a sequence of k_{ij} direct neighbour communications, where k_{ij} denotes the distance (number of links) between node i and node j. The amount of time required for sending a message of length M from node i to node j is given by $T = k_{ij}(\beta + M\tau)$ or, in the simplified terminology, k_{ij} steps are necessary for sending a message from node i to node j.

If the two communication partners are assumed to be arbitrarily chosen, the average distance in the network has to be minimum. In the worst case the sending node and the receiving node are at distance k (diameter of the network) from each other, therefore the minimum number of routing steps $\varrho_{1:1}$ in a given network of size N and degree d is given by the diameter of the network, which is in turn bounded by the minimum diameter:

$$\varrho_{1:1} = k \geq k_{min}$$

So we have to find a topology G which minimizes both, average distance and diameter for given degree and N.

The demands posed upon the routing algorithm have already been discussed in Section 2.3 (computational, shortest path routing and capability of failures).

[2] We use here the term *routing* as a generic name for all communication problems listed above.

6.1.2 Broadcasting

We will first assume that a node can only send and receive one message to (from) one of its neighbours within one step (single link availability). So the number of nodes knowing the message can at most double in each step and therefore the minimum number of routing steps is bounded by

$$\varrho_{1:N} = \lceil \log_2 N \rceil$$

If we assume, that a node can send one message to its d neighbours concurrently (multiple link availability), then the minimum number of routing steps is the diameter of the network:

$$\varrho_{1:N} = k \leq k_{min}$$

The routing strategy must find paths from the originator (sending node) to all other nodes and in order to prevent from redundancies each node should appear exactly once in this set of paths. This is in fact the definition of a spanning tree so the routing problem consists in embedding a spanning tree into the topology. The root node is the originator and each link corresponds to a call (sending a message from one node to one of tis neighbours). If we assume multiple link availability, the height of the tree determines the number of routing steps. In order to minimize the number of routing steps, we must find a spanning tree of minimum height.

For small graphs with low degree it should be possible find all spanning trees, since the number of possible spanning trees in graph G is given by $(-1)^{i+j} \|M\|$, the cofactor of a matrix M which is obtained from the negative adjacency matrix $-A$ of the graph by replacing each diagonal entry by the degree of the corresponding node [Wils 79]. For graphs with higher degree the number of possible spanning trees becomes too large, and it is necessary to find search procedures which take advantage of the special features of the topology (see Section 6.2).

6.1.3 Gossiping

Gossiping [Rich 88, Hede 88] is a problem which arises for instance in compute-aggregate-broadcast algorithms, where a node cannot proceed with its calculations (compute) until it has not received the messages from all other nodes and sent his message to all others (broadcast).

For the case of this complete message exchange we can find a lower bound by counting the maximum number of messages that any node can receive within one step. Every node has to receive $N-1$ messages totally. If a node can only send and receive a message over one link, then the minimum number of steps is simply given by the number of messages that any node has to receive, namely $N-1$.

If we assume that all nodes can send concurrently a message to all of their neighbours then any node can receive at most d different messages in the first step and at most $(d-1)(r-1)$ different messages in all the other $(r-1)$ steps, where r denotes the number of routing steps. In the optimum, no redundant messages are sent, e.g. the d messages received in the first step and the $d-1$ messages received in all following steps must sum up to the total number of $N-1$ messages that each single node has to receive. We can therefore establish the following relation between the minimum number of routing steps $\varrho_{N:N}$, the number of nodes in the network N and the node degree d:

$$N - 1 \quad = \quad d + (d-1)(\varrho_{N:N} - 1)$$

If we transform this equation with respect to $\varrho_{N:N}$ we obtain the minimum number of routing steps:

$$\varrho_{N:N} \quad = \quad \frac{N-2}{d-1}$$

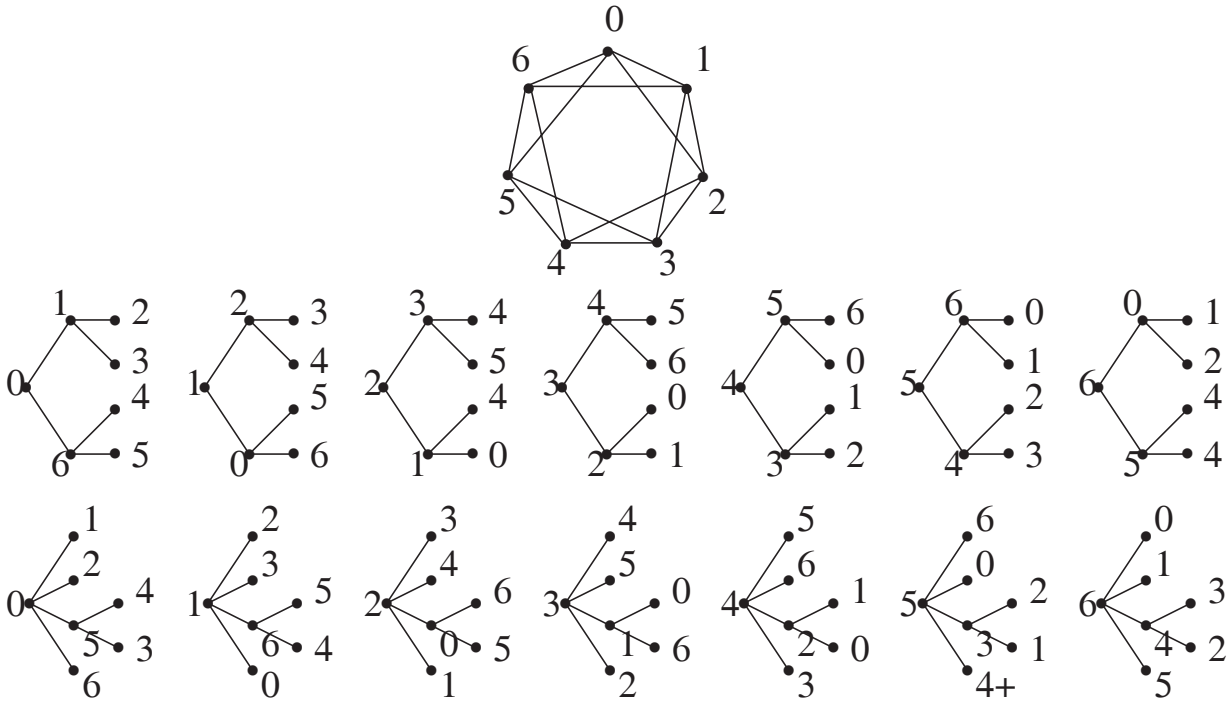


Figure 6.1: Gossiping with and without congestion

If a node can send and receive d different messages (one over each of its d links) the minimum number of steps is somewhat less and given by

$$\varrho_{N:N} = \frac{N}{d}$$

We can see from this equation that topologies with larger degree will perform better. But it is not possible to give a measure for a topology of given degree, by which we can tell whether it will be possible to perform gossiping in a minimum number of steps or not. We can only say, that irregular graphs will perform worse than regular ones, because a node with smaller node degree will not be able to send the d messages concurrently.

In gossiping every node has to broadcast its message. We can derive a routing strategy by applying the broadcast algorithm on every node. Note, that this algorithm can only reach the first bound $\varrho_{N:N} = \frac{N-2}{d-1}$. We have to find a forest of N spanning trees (every node is the root of exactly one tree) in the graph, but it is not assured that the minimum number of steps is reached since it can occur that a node would have to send two messages at the same time. Figure 6.1 shows the spanning trees for gossiping in a small example graph of order 7 and degree 3. It should be possible, to perform gossiping in a minimum number of 2 steps, therefore we have to find spanning trees of height 2. The graphs in the first row of figure 6.1 are a solution, which does not reach the optimum, because node 1 cannot send concurrently the messages from 0 and 2 (analogous for all other nodes in the second levels). The graphs in the second row of figure 6.1 (b) are an optimum solution. In order to cope with this problem, we must pose further restrictions upon the spanning tree ensuring that no node will be used twice within the same step:

1. The number of non leaf nodes in tree i is denoted by n_i.
2. A node in tree i is labelled (x, s), where x denotes the name (or address) of the node in the investigated topology and $s \in \{1, 2, \ldots, n_i\}$ denotes its number in the tree.

3. The root node of tree i is labelled $(i, 1)$.

4. No node has an s-label greater than its parent node.

5. No node has the same s-label in any tree.

6. The maximum number of non leaf nodes determines the number of steps necessary for complete message exchange: $\varrho = \max_i\{n_i\}$

In order to perform routing every node x has to store a vector $\vec{R}^x$ with ϱ elements where the value of the i-th element R_i^x denotes the tree in which x has label i. The routing algorithm is straightforward: every node x sends in step i the message from R_i^x. In a node symmetric graph the problem reduces to finding one spanning tree and restriction 5 can be ommitted.

In the next section we will apply this routing strategy to Extended Chordal Rings, where we will show, how it is possible to use the symmetry properties of ECRs for routing.

6.2 Topology dependent routing

6.2.1 Point-to-point-routing

We have already discussed various routing algorithms for point-to-point routing when reviewing the topologies (Chapter 3). We only want to note, that graphs on alphabets possess a very useful labelling scheme for nodes, which enables computational routing. Routing in Cayley Graphs is also interesting, because it corresponds to sorting the symbols of the source address into the destination address. In all tree based structures it should be possible to develop a routing algorithm based on routing in trees, with a little additional effort in order to make use of the topology specific properties (e. g. efficiently using the additional links in a Hypertree). The routing algorithm for Generalized Chordal Rings has the advantage that it is also applicable for CR's, DCR's and ECR's, but the disadvantage that it is not guaranteed, that the shortest path will always be found. Future research should try to find a routing algorithm for Extended Chordal Rings able to find the shortest path.

6.2.2 Broadcasting

As we have pointed out in the previous section, the problem of broadcasting consists in finding a spanning tree of minimum height in the topology, and the optimum is reached if the tree is of height k.

A broadcasting algorithm for Boolean n-cubes was found by Ho and Johnsson [Ho 86]. Based on this algorithm, El-Amawy and Latifi [El A 91] present an algorithm for broadcasting in Folded Hypercubes. In [Berm 91b] an algorithm for broadcasting in de Bruijn Graphs can be found and in [Bhuy 84] a braodcasting algorithm for Generalized Hypercubes is given.

6.2.3 Gossiping

In [Berm 91b] an algorithm for gossiping in de Bruijn Graphs can be found. Bertsekas et al. have studied the problem of gossiping in Boolean n-cubes [Bert 91] and in [Kund 91] the problem of gossiping in meshes and tori is analysed.

Gossiping in Extended Chordal Rings For gossiping in ECR's we assume that a node can concurrently send one message to all of its neighbours and simultaneously receiving messages from all of its neighbours. The minimum number of routing steps is therefore given by $\varrho_{N:N} = \frac{N-2}{d-1}$.

Since ECR's are p-symmetric, we have to find a spanning tree for every residue class. The spanning tree for an arbitrary node x is obtained by adding x to all nodes in the corresponding spanning tree. The procedure for finding the p spanning trees tries to minimize the number of

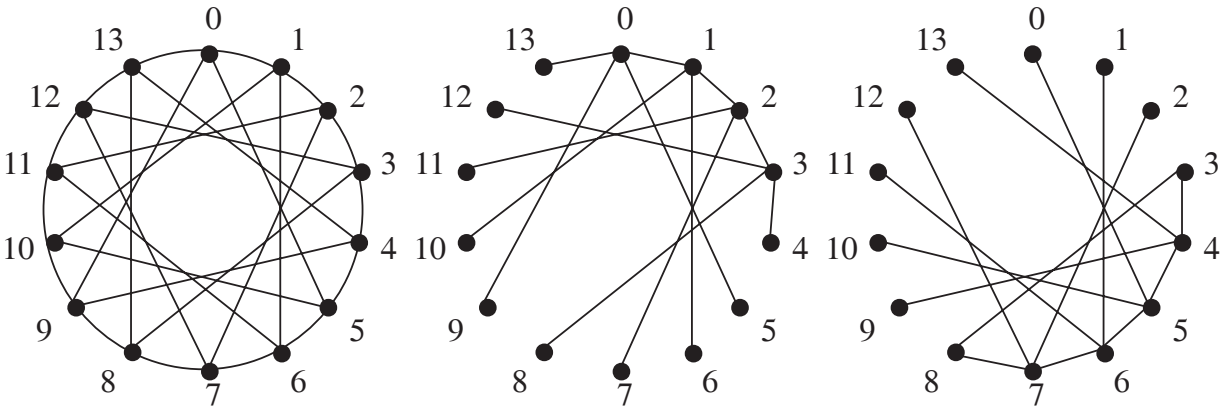


Figure 6.2: Spanning trees in an ECR

nonleaf nodes, which correspond to the number of routing steps (see Section 6.1), by maximizing the number of leaf nodes. We will start our explanation with the simple case of $p = 1$. We must start with the "class representative", who will send his message to all of its neighbours, which become its d children in the spanning tree (first routing step). In the next step one of these nodes must be chosen for further distribution of the message[3], and we will choose this node, which has the highest number of neighbours, not knowing the message from the representative so far. Those neighbours are added to the tree as children of the chosen node and become themselves candidates for passing on the message in the next step (in addition to all other leaf nodes so far in the tree). This procedure continues until all nodes appear in the tree.

We will illustrate this procedure with an ECR on 14 nodes with degree 4 and $W = (5; -5)$. Table 6.1) shows the construction of the spanning tree with node 0 as root node. In the first row of the table all nodes are listed. The next rows show the situations before each routing step, an entry in the row indicates, that this node is already in the tree and therefore able to send the message, the value of the entry indicates the number of adjacent nodes, that do not know the message so far (are not in the tree yet). At the beginning, only node 0 is able to send its message and 4 nodes will be added to the tree as its children. In the next step we can choose any of 0's children, because each of them will send the message to 3 new nodes. In such a situation, an arbitrary node will be chosen (assume for example node 1). After two further steps, all nodes appear in the tree (the last row has an entry 0 in each column). The spanning tree for any other node x is obtained by adding the node address x to all nodes in the tree. Figure 6.2 shows the spanning trees for nodes 0, and 4 as an example. So we have been able to complete gossiping in a minimum number of 4 routing steps.

Conjecture 1 *In any Extended Chordal Ring with degree $d = 4$, period $p = 1$ and a chord matrix $W = \begin{pmatrix} \varrho_{N:N} + 1 \\ -\varrho_{N:N} + 1 \end{pmatrix}$ an optimum spanning tree consists of $\varrho_{N:N}$ consecutive nodes on the ring as non leaf nodes and their adjacent nodes as leaves.*

The extension to higher periods requires the construction of p spanning trees. We can apply the same procedure, namely choosing always the node with the maximum number of adjacent nodes not reached so far in every period. But it is necessary to pose a further restriction in order to guarantee, that no node will be used at the same time in more than one tree, namely the chosen nodes must

[3]Since the spanning trees for all other nodes are obtained by addition of the node addresses, every node would need two nodes, sending its message in the next step, which is impossible.

node number														
0	1	2	3	4	5	6	7	8	9	10	11	12	13	sending node
4														0
0	3				3				3				3	1
0	0	3			1	2			2	1			3	2
0	0	0	3		1	0	2		2	0	1		3	3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	ready

Table 6.1: Gossiping in an $ECR(5;-5)$ on 14 nodes

belong to different residue classes. So it is necessary to choose the sending nodes simultaneously for all periods.

Chapter 7

Conclusions and Prospects

"acta est fabula"
Augustus

The major aims of this work were to give a comparative survey of static interconnection topologies, and to discuss their properties with respect to their use as interconnection topologies in message passing multi-computer systems. To this end it was necessary to recall relevant measures on graphs from graph theory, like for example the *average distance*, the *network diameter* or the *visit ratio*, and requirements from the parallel processing area, like the *reliability* or *extensibility*. Special emphasis has been given to present the construction rules for various graphs, because these seemed – along with the network characteristics – most relevant for interconnecting processing elements in reconfigurable multi-computer systems. Critical to applications in these kind of parallel systems is the possibility of exchanging local data among cooperating processing elements, so some space has also been devoted to the representation of the communication behaviour and the resulting *routing* demands.

According to the predefined measures it was possible in this work to point out those topological structures, that reach (or are close to) the theoretical optimum, showing that interconnection topologies currently used in parallel processing systems are often far from being 'best', and on the other hand that there are a lot of graphs meeting the specific requirements in parallel processing, which have not been used in this field so far. Hopefully this work is a contribution to bring these two disciplines closer to each other by either stimulating the investigation of graph theoretic results when deciding upon the interconnection network of a multi-computer system, or by directing the attention of researchers in graph theory to the demands (and restrictions) in parallel computation.

Appendix A

Derivations

A.1 Moore Bound and Minimum Diameter

An upper bound for the maximum number of nodes in a graph with given degree and diameter can be derived by the following considerations: For one node it is possible to reach at most d nodes in one step, $d(d-1)$ nodes in the second step and so on. If we draw this reachability graph we will a maximum tree of height k and degree d otherwise, and the total number of nodes in this tree gives a bound for the maximum number of nodes in the network. If $d = 2$ the tree is degenerated (a line) and the maximum number of nodes is given by

$$N_{max} = 1 + 2k$$

otherwise ($d > 2$) the maximum number of nodes is given by

$$\begin{aligned} N_{max} &= 1 + d + d(d-1) + d(d-1)^2 + \cdots + d(d-1)^{h-1} = \\ &= 1 + \sum_{i=1}^{k} d^i (d-1)^{i-1} = \\ &= \frac{d(d-1)^k - 2}{d-2} \end{aligned}$$

The lower bound for diameter is obtained by transforming those equations:

$$\begin{aligned} N &= 1 + 2k_{min} \\ k_{min} &= (N-1)/2 \end{aligned}$$

and for $d > 2$ we have

$$\begin{aligned} N &= \frac{d(d-1)^{k_{min}} - 2}{d-2} \\ N(d-2) + 2 &= d(d-1)^{k_{min}} \\ \log_{d-1}(N(d-2)+2)/d) &= \log_{d-1}(d-1)^{k_{min}} \\ k_{min} &= \log_{d-1}(N(d-2)+2)/d \end{aligned}$$

A.2 Bound on Average Distance

Analogous considerations will lead to a lower bound for average distance under given N and d. We will again consider a maximum tree and calculate the average distance for the root node, where the height of the tree is given by the minimum diameter for N and d and we have to consider the fact that the last level will not be full if $N < N_{max}$:

$$\begin{aligned}\mu_{min} &= (1 * d + 2 * d(d-1) + 3 * d(d-1)^2 + \cdots + \\ &\quad + k_{min} * (d(d-1)^{k_{min}-1} - (N_{max} - N))/(N-1) = \\ &= \frac{\sum_{i=1}^{k_{min}} i\, d\, (d-1)^{i-1} - k_{min}(N_{max} - N)}{N-1}\end{aligned}$$

A.3 Proofs

We want to show, that $N_{Friedman}(d,k) < N_{Korn}(d,k)$ only if k is odd:

$$\begin{aligned}N_{Friedman}(d,k) &= \frac{2(d(d-1)^{\lfloor k/2 \rfloor} - d)}{d-2} \\ N_{Korn}(d,k) &= \frac{2(2(d-1)^{\lfloor (k+1)/2 \rfloor} - d)}{d-2}\end{aligned}$$

Since k is odd we obtain

$$\begin{aligned}\frac{2(d(d-1)^{(k-1)/2} - d)}{d-2} &< \frac{2(2(d-1)^{(k+1)/2} - d)}{d-2} \\ d(d-1)^{(k-1)/2} &< 2(d-1)^{(k+1)/2} \\ \frac{d}{2} &< \frac{(d-1)^{(k+1)/2}}{(d-1)^{(k-1)/2}} \\ \frac{d}{2} &< d-1 \\ d &> 2\end{aligned}$$

If the diameter is even, we obtain

$$\begin{aligned}\frac{2(d(d-1)^{k/2} - d)}{d-2} &> \frac{2(2(d-1)^{k/2} - d)}{d-2} \\ d(d-1)^{k/2} &> 2(d-1)^{k/2} \\ \frac{d}{2} &> \frac{(d-1)^{k/2}}{(d-1)^{k/2}} \\ d &> 2\end{aligned}$$

So, $N_{Friedman} > N_{Korn}$ if k is even and $N_{Korn} > N_{Friedman}$ if k is odd and $d > 2$, which is true for all K- and F-trees.

A.4 Average Distance

Line The topology is irregular but we only have to calculate μ_i for nodes $i \leq (N+1)/2$ because of symmetry.

Average distance for node i with $1 \leq i \leq \frac{N+1}{2}$:

$$\begin{aligned}
\mu_i &= \frac{1}{N-1}(\sum_{j=1}^{i-1} j + \sum_{j=1}^{N-i} j) = \\
&= \frac{1}{N-1}(\frac{i(i-1)}{2} + \frac{(N-i)(N-i+1)}{2}) = \\
&= \frac{i^2 - i + N^2 - Ni - Ni + i^2 + N - i}{2(N-1)} \\
&= \frac{N^2 + N - 2i(N+1) + 2i^2}{2(N-1)}
\end{aligned}$$

For even N:

$$\begin{aligned}
\mu &= \frac{1}{N} 2 \sum_{i=1}^{N/2} \mu_i = \\
&= \frac{2}{N} \frac{1}{2(N-1)} ((N^2+N)N/2 + 2\frac{N(N/2+1)(N+1)}{6} + 2(N+1)N(N+1)/2) = \\
&= \frac{N+1}{3}
\end{aligned}$$

For odd N:

$$\begin{aligned}
\mu &= \frac{1}{N}(2 \sum_{i=1}^{(N-1)/2} \mu_i + \mu_{(N+1)/2}) = \\
&= \frac{N+1}{3}
\end{aligned}$$

Ring Since the topology is regular and node symmetric it is sufficient to calculate the average distance for a single point in the ring where we only have to distinguish between an even or an odd number of nodes.

For even N:

$$\begin{aligned}
\mu &= \frac{1}{N-1}(\sum_{i=1}^{(N-2)/2} 2i + N/2) = \\
&= \frac{1}{N-1}(\frac{N}{2} + \frac{(N-2)}{2}\frac{N}{2}) = \\
&= \frac{N^2}{4(N-1)}
\end{aligned}$$

For odd N:

$$\begin{aligned}\mu &= \frac{1}{N-1}(\sum_{i=1}^{(N-1)/2} 2i) = \\ &= \frac{2}{N-1}(N-1)(N+1)/8 \\ &= \frac{N+1}{4}\end{aligned}$$

Star The topology is irregular, we therefore have to calculate μ_i for the root and only for one leaf and can calculate the average distance as follows:

$$\begin{aligned}\mu_{root} &= 1 \\ \mu_{leaf} &= \frac{1*1+2*(N-2)}{N-1} = \frac{2N-3}{N-1} \\ \mu &= \frac{\mu_{root}+(N-1)\mu_{leaf}}{N} = \frac{2N-2}{N}\end{aligned}$$

Appendix B

Symbols and abbreviations

B.1 Mathematical Symbols

$\oplus_N$ ..... addition modulo N
$\ominus_N$ ..... subtraction modulo N
$\lfloor x \rfloor$ ..... largest integer not larger than x
$\lceil x \rceil$ ..... smallest integer not less than x

B.2 Abbreviations for Measures

$\mathcal{C}$ ..... connectedness
d ..... degree
γ_{link} ..... most-loaded link load
γ_{node} ..... worst through routing load
k ..... diameter
κ ..... connectivity
L ..... total number of links
λ ..... cohesion
μ ..... average distance
μ_{norm} ..... normalized average distance
N ..... total number of nodes
ν ..... narrowness
$\mathcal{R}$ ..... reliability function
ρ ..... persistence
$\mathcal{U}$ ..... unreliability function
V ..... visit ration

B.3 Boolean Operations on Graphs

$G_1 \times G_2$ ..... Cartesian Product
$G_1 \star G_2$ ..... Generalized Product
$G_1 \wedge G_2$ ..... Conjunction
$G_1[G_2]$ ..... Composition
$G_1 \vee G_2$ ..... Disjunction
$G_1 \diamond G_2$ ..... Symmetric Difference
$G_1 \setminus G_2$ ..... Rejection

B.4 Abbreviations for Topologies

$BG(b,n)$ ..... de Bruijn Graph with words of length n in a radix b representation
$BQ(n,s)$ ..... Banyan Hypercube of dimension n with h levels and indegree s
$BSG(n)$ ..... Bubblesort Graph as a permutation of n symbols
$C(N)$ ..... cycle (ring) on N nodes
$C1(d,m)$ ..... C1 Graph of degree d and m nodes in a ring
$CCC(n)$ ..... Cube Connected Cycles of diemnsion n
$CR(w)$ ..... Chordal Ring with chord w
$\mathcal{D}T^m(B,h,p,K)$ ..... $\mathcal{D}_{\mathcal{K}}$-tree made of m rooted subtrees with degree B and height h and all leaves are connected as a graph on p nodes with degree $B-1$ and diameter K
$DCR(w_1,w_2)$ ..... Double Chordal Ring with chord lengths w_1 and w_2
$ECR(W)$ ..... Extended Chordal Ring with chord matrix W
$FQ(n)$ ..... Folded Hypercube of dimension n
$FT(d,h)$ ..... F-tree of degree d and height h
$GCR(W)$ ..... Generalized Chordal Ring with chord matrix W
$GHC(\vec{m})$ ..... Generalized Hypercube with m_i nodes in dimension i
$GQ(R,n)$ ..... Generalized boolean n-cube
$HC(W,n)$ ..... W-ary n-cube
$Hnet(B,e,G,h)$ ..... Hypernet
$HT(h)$ ..... Hypertree of height h
$K(N)$ ..... Complete graph on N nodes
$KG(b,n)$ ..... Kautz Graph
$KT(d,h)$ ..... K-tree of degree d and height h
$L(N)$ ..... line with N nodes
$M(W,n)$ ..... Mesh of dimension n and radix W
$MDMW(N)$ ..... minimum distance mesh with wrap around links on N nodes
$MG(n)$ ..... Moebius Graph
$MQ(n)$ ..... multiply-twisted cube of dimension n
$MTS(m,B,h))$ ..... Multitree with m subtrees, B branches, and height h
$NBG(d,l,m_1,m_2)$ ..... Nonidentical Blocking graph of degree d with l levels
$NHG(d,l,m_1,m_2)$ ..... Nonidentical Hinging graph of degree d with l levels
$OG(d)$ ..... Odd Graph with diameter d
$OT(h)$ ..... Ortogonal tree network of height h
$Q(n)$ ..... Boolean n-cube
$PG(n)$ ..... Pancake Graph as a permutation of n symbols
$S(N)$ ..... star on N nodes
$SG(n)$ ..... Star Graph according to Akers as a permutation of n symbols
$SOS^n(n)$ ..... Shift n and Swap graph
$SOS^{n-1}(n)$ ..... Shift $n-1$ and Swap graph
$ST(h)$ ..... Sneptree of height h
$T(B,H)$ ..... tree with B branches and height h
$T^{full}(B,H)$ ..... complete tree with B branches and height h
$T^{max}(B,H)$ ..... maximum tree with B branches and height h
$Tor(m,r,p_{up},p_{down})$ ..... Torus according to von Conta
$TQ(n)$ ..... Twisted Cube of dimension n according to Esfahanian

List of Figures

List of Tables

Bibliography

[Adam 82] G. B. Adams and H. J. Siegel. "The Extra-Stage Cube: A Fault-Tolerant Interconnection Network for Supersystems". *IEEE Transactions on Computers*, Vol. C-31, pp. 443–454, May 1982.

[Adam 87] G. B. Adams, D. P. Agrawal, and H. J. Siegel. "A Survey and Comparison of Fault-Tolerant Multistage Interconnection Network". *Computer*, pp. 14–27, June 1987.

[Agra 78] D. P. Agrawal, T. Y. Feng, and C. L. Wu. "A survey of communication processor systems". In: *Proceedings COMPSAC*, pp. 668–673, November 1978.

[Agra 83] D. P. Agrawal. "Graph theoretic analysis and design of multistage interconnection networks". *IEEE Transactions on Computers*, Vol. C-32, pp. 637–648, July 1983.

[Agra 86] D. P. Agrawal, V. K. Janakiram, and G. C. Pathak. "Evaluating the Performance of Multicomputer Configurations". *IEEE Computer*, pp. 23–37, May 1986.

[Aker 65] S. B. Akers. "On the Construction of (d, k) Graphs". *IEEE Transactions on Electronic Computers*, Vol. EC-14, p. 488, June 1965.

[Aker 84] S. B. Akers and B. Krishnamurthy. "Group Graphs as Interconnection Networks". In: *Proceedings 14th International Conference on Fault Tolerant Computing*, pp. 422–427, 1984.

[Aker 86] S. B. Akers and B. Krishnamurthy. "A Group Theoretic Model for Symmetric Interconnection Networks". In: K. Hwang, S. M. Jacobs, and E. E. Swartzlander, Eds., *Proceedings of International Conference on Parallel Processing*, pp. 216–223, IEEE, August 1986.

[Aker 87] S. B. Akers, D. Harel, and B. Krishnamurthy. "The Star Graph: An Attractive Alternative to the n-Cube". In: S. K. Sahni, Ed., *Proceedings of International Conference on Parallel Processing*, pp. 393–400, IEEE, August 1987.

[Akl 90] S. G. Akl, J. Duprat, and A. Ferreira. "Building hamiltonian circuits and paths on Star Graphs". Tech. Rep. 90-22, LIP, Ecole Normale Supériore de Lyon, 1990. October.

[Aleg 86] I. Alegre, M. A. Fiol, and J. L. A. Yebra. "Some large graphs with given degree and diameter". *Journal of Graph Theory*, Vol. 10, pp. 219–224, 1986.

[Ande 75] G. A. Anderson and E. D. Jensen. "Computer Interconnection Structures; Taxonomy, Characteristics, and Examples". *ACM Computing Surveys*, Vol. 7, No. 4, pp. 197–212, December 1975.

[Arde 78] B. Arden and H. Lee. "A multi-tree structured Network". In: *Proceedings 17th COMPCON*, pp. 201–210, September 1978.

[Arde 81] B. W. Arden and H. Lee. "Analysis of Chordal Ring network". *IEEE Transactions on Computers*, Vol. C-30, No. 4, pp. 25–29, April 1981.

[Arde 89] B. W. Arden and K. W. Tang. "Representation and Routing of Cayley Graphs". Tech. Rep. EL-89-02, Dep. of Electrical Engineering, Univerity of Rochester, 1989.

[Arde 90] B. W. Arden and K.-M. W. Tang. "Routing for Generalized Chordal Rings". In: *Proceedings of the Eighteenth Annual Computer Science Conference*, ACM, February 1990.

[Arms 81] J. R. Armstrong and F. G. Gray. "Fault diagnosis in a Boolean n-cube array of microprocessors". *IEEE Transactions on Computers*, Vol. C-30, No. 8, pp. 587–590, August 1981.

[Arru 90] A. Arruabarrena, R. Beivide, E. Herrada, J. L. Balcàcar, and C. Izu. "An Optimal Topology for Multicomputer Systems Based on a Mesh of Transputers". *Occam User Group Letters*, No. 12, pp. 31–39, January 1990.

[Atha 88] W. C. Athas and C. L. Seitz. "Multicomputers: message-passing concurrent computers". *IEEE Computer*, No. , pp. 9–24, August 1988. IEEE.

[Bail 88] C. F. Baillie. "Comparing shared and distributed memory computers". *Parallel Computing*, No. 8, pp. 101–110, 1988. North Holland.

[Baue 85] D. Bauer, F. Boesch, C. Suffel, and R. Tindell. "Combinatorial optimization problems in the analysis and design of probabilistic networks". *Networks*, Vol. 15, pp. 257–271, 1985.

[Beiv 87a] R. Beivide, E. Herrada, J. L. Balcazar, and J. Labarta. "Optimized mesh-connected networks for SIMD and MIMD architectures". In: *Proceedings of the 14th International Symposium on Computer Architecture*, pp. 163–170, June 1987.

[Beiv 87b] R. Beivide, V. Vinals, and C. Rodriguez. "Optimal Topologies with Four Bidirectional Links and its Applications". In: *Proceedings of The IASTED: International Symposium on Applied Informatics*, pp. 66–69, February 1987.

[Berg 62] C. Berge. *The Theory of Graphs and its Applications*. Methuen, London, 1962.

[Berm 82a] J. C. Bermond, C. Delorme, and G. Farhi. "Large graphs with given degree and diameter 111". *Ann. Discrete Math.*, Vol. 13, pp. 23–32, 1982. in Graph Theory.

[Berm 82b] J.-C. Bermond, C. Delorme, and J.-J. Quisquater. "Tables of large graphs with given degree and diameter". *Information Processing Letters*, Vol. 15, pp. 10–13, August 1982.

[Berm 83] J. C. Bermond, J. B. M. Paoli, and C. Peyrat. "Graphs and intercommunication networks: diameter and vulnerability". *Surveys in Combinatorics , Lond. Math. Soc. Lecture Notes*, Vol. 82, pp. 1–30, 1983. (LLoyd, E. K., Ed.).

[Berm 84] J. C. Bermond, C. Delorme, and G. Farhi. "Large graphs with given degree and diameter 11". *Journal of Combinatorial Theory*, Vol. Series B 36, pp. 32–48, 1984.

[Berm 86] J.-C. Bermond, C. Delorme, and J.-J. Quisquater. "Strategies for Interconnection networks: Some Methods from Graph Theory". *Journal of Parallel and Distributed Computing*, Vol. 3, No. 4, pp. 433–449, December 1986.

[Berm 91a] J.-C. Bermond. "Table of large (d, Δ) graphs". 1991. Personal Communication.

[Berm 91b] J.-C. Bermond and P. Fraigniaud. "Broadcasting and Gossiping in de Bruijn Networks". Tech. Rep., LIP, Ecole Normale Supériore de Lyon, 1991. March.

[Bert 91] D. P. Bertsekas, C. Özveren, G. D. Stamoulis, P. Tseng, and J. N. Tsitsiklis. "Optimal Communication Algorithms for Hypercubes". *Journal of Parallel and Distributed Computing*, Vol. 11, pp. 263–275, 1991.

[Bhat 82] K. V. S. Bhat. "Fault Diagnosis in Hypercube Connected Arrays of Processors". In: *Proceedings of the Third International Conference on Distributed Computing Systems*, pp. 318–321, IEEE, October 1982.

[Bhuy 82] L. N. Bhuyan and D. P. Agrawal. "A General Class of Processor Interconnection Strategies". In: *Proceedings of the 9th Annual Symposium on Computer Architecture*, pp. 90–98, April 1982.

[Bhuy 83] L. N. Bhuyan and D. P. Agrawal. "Design and Performance of Generalized Interconnection Networks". *IEEE Transactions on Computers*, Vol. C-32, December 1983.

[Bhuy 84] L. N. Bhuyan and D. P. Agrawal. "Generalized Hypercube and Hyperbus Structures for a Computer Network". *IEEE Transactions on Computers*, Vol. C-33, pp. 323–333, April 1984.

[Bhuy 87] L. N. Bhuyan. "Interconnection Networks for Parallel and Distributed Processing". *IEEE Computer*, Vol. 20, No. 6, pp. 9–12, June 1987.

[Boes 70] F. T. Boesch and R. E. Thomas. "On Graphs of Invulnerable Communication Nets". *IEEE Transactions on Circuit Theory*, Vol. CT-17, pp. 183–191, May 1970.

[Boes 81] F. T. Boesch, F. Harary, and J. A. Kabell. "Graphs as models of communication network vulnerability: Connectivity and Persistence". *Networks*, Vol. 11, No. 1, pp. 57–63, 1981.

[Boes 85] F. T. Boesch and J.-F. Wang. "Reliable Circulant Networks with Minimum Transmission Delay". *IEEE Transactions on Circuits and Systems*, Vol. CAS-32, pp. 1286–1291, December 1985.

[Boll 78] B. Bollobás. *Extremal graph theorie*. Academic press, 1978.

[Boll 85] B. Bollobás. *Random Graphs*. Academic Press, 1985.

[Brua 67] R. A. Brualdi. "Kronecker Products of fully indecomposable matrices and of ultrastrong digraphs". *Journal of Combinatorial Theory*, Vol. 2, pp. 135–139, 1967.

[Brui 46] N. F. de Bruijn. "A combinatorial problem". In: *Proceedings Vol.49, part 20*, Koninklijke Nederlands, Academe van Wetenschappen, 1946.

[Carl 85] G. E. Carlsson, J. E. Cruthirds, H. B. Sexton, and C. G. Wright. "Interconnection Networks Based on a Generalization of Cube-Connected Cycles". *IEEE Transactions on Computers*, Vol. C-34, pp. 769–772, August 1985.

[Catt 79] K. W. Cattermole. "Graph Theory and Communication networks". In: *Applications of Graph Theory*, pp. 17–57, Academic Press, 1979.

[Cayl 78] A. Cayley. "The Theroy of Groups; Graphical Representation". *American Journal of Mathematics*, Vol. 1, pp. 175–176, 1878.

[Chen 90] M.-S. Chen and K. G. Shin. "Adaptive Fault-Tolerant Routing in Hypercube Multicomputers". *IEEE Transactions on Computers*, Vol. C-39, No. 12, pp. 1406–1416, December 1990.

[Cont 83] C. v. Conta. "Torus and other Networks as Communication Networks with up to some 100 points". *IEEE Transactions on Computers*, Vol. C-32, pp. 657–666, July 1983.

[Dall 86] W. J. Dally and C. L. Seitz. "The torus routing chip". *Distributed Computing*, Vol. 1, pp. 187–196, 1986. Springer Verlag.

[Dall 90] W. J. Dally. "Performance Analysis of k-ary n-cube". *IEEE Transactions on Computers*, Vol. C-39, No. 6, pp. 775–785, June 1990.

[Delo 84] C. Delorme and G. Fahri. "Large Graphs with given degree and diameter". *IEEE Transactions on Computers*, Vol. C-33, pp. 857–860, September 1984.

[Delo 85] C. Delorme. "Large bipartite graphs with given degree and diameter". *J. Graph Theory*, Vol. 8, No. 3, pp. 325–334, 1985.

[Desp 78] A. M. Despain and D. A. Patterson. "X-Tree: A Tree Structured Multiprocessor Computer Architecture". In: *Proceedings of the 5th Annual Symposium on Computer Architecture*, pp. 144–151, April 1978.

[Doty 82] K. W. Doty. "Large regular interconnection networks". In: *Proc Third International Conference on Distributed Computing Systems*, pp. 312–317, IEEE, October 1982.

[Doty 84] K. W. Doty. "New Designs for Dense Processor Interconnection Networks". *IEEE Transactions on Computers*, Vol. C-33, No. 5, pp. 447–450, May 1984.

[Dunc 90] R. Duncan. "A Survey of Parallel Computer Architectures". *IEEE Computer*, pp. 5–16, February 1990.

[Efe 88] K. Efe, P. Blackwell, T. Shiau, and W. Slough. "A Reduced Diameter Interconnection Network". In: *Frontiers '88, The 2nd Symposium on the Frontiers of Massively Parallel Computation*, pp. 471–474, IEEE, October 1988.

[Efe 91] K. Efe. "A Variation on the Hypercube with Lower Diameter". *IEEE Transactions on Computers*, Vol. 40, No. 11, p. 1312, November 1991.

[El A 91] A. El-Amawy and S. Latifi. "Properties and Performance of Folded Hypercubes". *IEEE Transactions on Parallel and Distributed Systems*, Vol. 2, pp. 31–42, January 1991.

[Elsp 64] B. Elspas. "Topological Constraints on Interconnection-limited Logic". *Switching Circuit Theory and Logical Design*, Vol. S-164, No. 5, pp. 133–147, October 1964.

[Ensl 80] P. H. Enslow, Ed. *Multiprocessors and Parallel Processing.* John Wiley & Sons, 1980.

[Erdo 80] P. Erdös, S. Fajtlowicz, and A. J. Hoffman. "Maximum degree in graphs of diameter 2". *Networks*, Vol. 10, No. 1, pp. 87–90, 1980.

[Esfa 88] A.-H. Esfahanian, L. M. Ni, and B. E. Sagan. "On enhancing hypercube multiprocessors". In: *Proceedings 1988 International Conference on Parallel Processing*, pp. 86–98, 1988.

[Exoo 82] G. Exoo. "On a measure of communication network vulnerability". *Networks*, Vol. 12, No. 1, pp. 405–409, 1982.

[Feng 81] T. Feng. "A Survey of Interconnection Networks". *IEEE Computer*, pp. 12–27, December 1981.

[Fers 91a] A. Ferscha. "Private communication". April 1991.

[Fers 91b] A. Ferscha and G. Kotsis. "Eliminating Routing overheads in Neural Network simulation using chordal ring interconnection topologies". Tech. Rep. ACPC/TR 91-8, Austrian Center for Parallel Computation, 1991. February.

[Fink 80] R. A. Finkel and M. H. Solomon. "Processor Interconnection Strategies". *IEEE Transactions on Computers*, Vol. C-29, pp. 360–371, May 1980.

[Fink 81] R. A. Finkel and M. H. Solomon. "The Lens Interconnection Strategy". *IEEE Transactions on Computers*, Vol. C-30, pp. 960–965, December 1981.

[Flyn 66] M. J. Flynn. "Very high-speed computing systems". In: *Proceedings of the IEEE, Vol. 54*, pp. 1901–1909, IEEE, 1966.

[Ford 62] L. R. Ford and D. R. Fulkerson. *Flows in Networks.* Princeton University Press, Princeton, 1962.

[Fost 32] R. M. Foster. "Geometrical circuits of electrical networks". *Trans. Amer. Inst. Elec. Engrs.*, Vol. 51, pp. 399–404, 1932.

[Fran 69] H. Frank. "Maximally reliable node weighted graphs". In: *Proc. 3rd Annual Princeton Conference on Information Sciences and Systems*, pp. 1–6, March 1969.

[Fran 71] H. Frank and I. T. Frisch. *Communication, Transmission and Transportation networks.* Addison-Wesley, 1971.

[Frec 70] J. B. Frechen. "Graph connectivity algorithm". In: *Proceedings of the International Conference Combinatorial Mathematics*, New York Acad. Sci., April 1970.

[Frie 66] H. D. Friedman. "A Design for (d, k) Graphs". *IEEE Transactions on Electronic Computers*, Vol. EC-15, pp. 253–254, April 1966.

[Fris 67] I. T. Frisch. "Analysis of the vulnerability of communication nets". In: *Proc. 1st Annual Princeton Conference on Information Sciences and Systems*, pp. 188–192, 1967.

[Fris 69] I. T. Frisch. "Flow variation in multiple min-cut calculations". *Journal of the Franklin Institute*, Vol. 287, pp. 61–72, January 1969.

[Fure 90] Z. Füredi. "Graphs of diameter 3 with the minimum number of edges". *Graphs and Combinatorics*, Vol. 6, pp. 333–337, 1990.

[Gate 79] W. H. Gates and C. H. Papadimitriou. "Bounds for sorting by prefix reversal". *Ann. Discrete Math.*, Vol. 27, pp. 47–57, 1979.

[Gilo 86] W. K. Giloi. *Interconnection Networks for massively Parallel Computer Systems*, pp. 321–348. *Lecture Notes in Computer Science: Future Parallel Computers*, Springer Verlag, June 1986.

[Goke 73] L. R. Goke and G. J. Lipovski. "Banyan networks for partitioning multiprocessor systems". In: *Proceedings Annual Symposium on Computer Architecture*, pp. 21–28, December 1973.

[Gome 85] J. Gómez and M. A. Fiol. "Dense compound graphs". In: *Proceedings Tenth British Combinatorial Conf.*, pp. 211–237, Glasgow, 1985. Ars Combin. 20-A.

[Gomo 61] R. E. Gomory and T. C. Hu. "Multiterminal network flows". *SIAM Journal of Applied Mathematics*, Vol. 9, pp. 551–570, December 1961.

[Good 81] J. Goodman and C.H.Sequin. "Hypertree: A Multiprocessor Interconnection Topology". *IEEE Transactions on Computers*, Vol. C-30, No. 12, pp. 923–933, December 1981.

[Hara 59] F. Harary. "On the group of the composition of two graphs". *Duke Math.J.*, Vol. 26, pp. 29–34, 1959.

[Hara 66] F. Harary and C. A. Trauth. "Connectedness of products of two directed graphs". *SIAM Journal of Applied Math.*, Vol. 14, pp. 250–254, 1966.

[Hara 67] F. Harary and G. Wilcox. "Boolean Operations on graphs". *Math. Scand.*, Vol. 20, pp. 41–51, 1967.

[Hara 72] F. Harary. *Graph Theory*. Addison-Wesley, 1972.

[Harr 79] J. Harris and D. Smith. "Simulation experiments of a tree organized multicomputer". In: *Proceedings of the 6th Annual Symposium on Computer Architecture*, pp. 83–89, IEEE, 1979.

[Hawk 85] L. W. Hawkes. "A Regular Fault-Tolerant Architecture for Interconnection Networks". *IEEE Transactions on Computers*, Vol. C-34, pp. 677–680, July 1985.

[Hede 88] S. M. Hedetniemi, S. T. Hedetniemi, and A. L. Liestman. "A Survey of gossiping and Broadcasting in Communication Networks". *Networks*, Vol. 18, No. 4, pp. 319–349, 1988.

[Hein 89] L. J. Heinrich and F. Roithmayr. *Wirtschaftsinformatik-Lexikon*. Oldenbourg, 3. auflage Ed., 1989.

[Hilb 87] P. A. J. Hilbers, M. R. J. Koopman, and J. L. A. van de Snepscheut. *The Twisted Cube*, pp. 152–159. *Lecture Notes in Computer Science: PARLE*, Springer Verlag, June 1987.

[Ho 86] C. T. Ho and S. L. Johnsson. "Distributed Routing algorithms for broadcasting and personalized Communication in hypercubes". In: K. Hwang, S. M. Jacobs, and E. E. Swartzlander, Eds., *Proceedings International Conference on Parallel Processing*, pp. 640–648, IEEE, August 1986.

[Hobb 88] L. C. Hobbs, D. J. Theis, J. Trimble, H. Titus, and I. Highberg, Eds. *Parallel Processor Systems, Technologies , and Applications*. Spartan Books, 1988.

[Hoff 60] A. J. Hoffman and R. R. Singleton. "On Moore Graphs with Diameters 2 and 3". *IBM Journal of Research and Development*, Vol. 4, pp. 497–504, November 1960.

[Horo 79] E. Horowitz and A. Zorat. "A divide and conquer computer". Tech. Rep., Dep. Comput. Sci., Univ. of Southern California, 1979. July.

[Horo 81] E. Horowitz and A. Zorat. "The Binary Tree as an Interconnection Network: Applications to Multiprocessor Systems and VLSI". *IEEE Transactions on Computers*, Vol. C-30, pp. 247–253, April 1981.

[Huan 88] T. C. Huang, J. F. Wang, C. S. Yang, and J. Y. Lee. "The number of spanning trees of the generalized Boolean n-cube network". *Journal Comput. Math. Appl.*, Vol. 16, No. 9, pp. 715–725, 1988.

[Huan 89] T.-C. Huang, J.-F. Wang, C.-S. Yang, and J.-Y. Lee. "Graph theoretic characterization and reliability of the generalized Boolean n-cube network". *Parallel Computing*, Vol. 12, pp. 375–385, 1989. North Holland.

[Hwan 81] C. L. Hwang, F. A. Tillman, and M. H. Lee. "System-reliability evaluation techniques for complex/large systems". *IEEE Trans. on Reliability*, Vol. R-30, pp. 416–422, December 1981.

[Hwan 84] K. Hwang and F. A. Briggs. *Computer Architecture and Parallel Processing*. McGraw-Hill, 1984.

[Hwan 87] K. Hwang and J. Ghosh. "Hypernet: A Communication-Efficient Architecture for Constructing Massively Parallel Computers". *IEEE Transactions on Computers*, Vol. C-36, No. 12, pp. 1450–1466, December 1987.

[Imas 81] M. Imase and M. Itoh. "Design to Minimize Diameter on Building-block Network". *IEEE Transactions on Computers*, Vol. C-30, pp. 439–442, June 1981.

[Imas 83] M. Imase and M. Itoh. "A Design for directed Graphs with minimum Diameter". *IEEE Transactions on Computers*, Vol. C-32, pp. 782–784, August 1983.

[Jain 88] B. N. Jain and S. K. Tripathi. "An Analysis of Cube-Connected Cycles and Circular Shuffle Networks for Parallel Computation". *Journal of Parallel and Distributed Computing*, No. 5, pp. 741–754, May 1988.

[Jin 84] L. Jin and Y. yuan Yang. "Analysis of Double-chordal Ring Interconnection Networks". In: *The First International Conference on Computers and Applications*, pp. 248–255, IEEE, June 1984.

[Kaut 68] W. H. Kautz. "Bounds on directed (d, k) graphs". *Theory of Cellular Logic Networks and Machines*, pp. 20–28, 1968. AFCRL-68-0668, SRI Project 7258, final report.

[Kelm 67] A. K. Kel'mans. "Connectivity of probabilistic networks". *Automat. Remote Control*, No. 3, pp. 444–460, 1967.

[Klei 69] D. J. Kleitman. "Methods for Investigating Connectivity of Large Graphs". *IEEE Transactions on Circuit Theory*, Vol. CT-16, pp. 232–233, May 1969.

[Korn 67] I. Korn. "On (d, k) Graphs". *IEEE Transactions on Electronic Computers*, Vol. EC-16, p. 90, February 1967.

[Kund 91] M. Kunde and T. Tensi. "(k-k) Routing on Multidimensional Mesh-Connected Arrays". *Journal of Parallel and Distributed Computing*, Vol. 11, pp. 146–155, 1991.

[Lela 81] W. E. Leland, R. Finkel, L. Qiao, M. Solomon, and L. Uhr. "High Density graphs for Processor Interconnection". *Information Processing Letters*, No. 12, pp. 117–120, 1981.

[Lela 82] W. E. Leland and M. H. Solomon. "Dense trivalent Graphs for Processor Interconnection". *IEEE Transactions on Computers*, Vol. C-31, pp. 219–222, March 1982.

[Leng 82] T. Lengauer. "Cube-Connected-Cycles — Shuffle-Exchange graph". *Informatik-Spektrum*, No. 5, pp. 192–194, 1982.

[Li 86] P. P. Li and A. J. Martin. "The Sneptree — a versatile interconnection Network". In: K. Hwang, S. M. Jacobs, and E. E. Swartzlander, Eds., *Proceedings of the International Conference on Parallel Processing*, pp. 20–27, IEEE, August 1986.

[Mart 90] A. J. Martin and J. L. A. van de Snepscheut. "An Interconnection Network for Distributive Recursive Computations". *IEEE Transactions on Computers*, Vol. C-39, No. 11, pp. 1393–1395, November 1990.

[McAn 63] M. H. McAndrew. "On the product of directed graphs". In: *Proceedings Amer. Math. Soc.*, pp. 600–606, 1963. No. 14.

[Memm 82] G. Memmi and Y. Raillard. "Some new results about the (d,k) graph problem". *IEEE Transactions on Computers*, Vol. C-31, pp. 784–791, August 1982.

[Nath 83] D. Nath, S. N. Maheshwari, and P. C. P. Bhatt. "Efficient VLSI networks for Parallel Processing based on orthogonal trees". *IEEE Transactions on Computers*, Vol. C-32, No. 6, pp. 569–581, June 1983.

[Norm 88] M. G. Norman. "A Genetic Approach to Topology Optimisation for Multiprocessor Architectures". 1988. Edinburgh Preprint 88/451.

[Norm 89] M. G. Norman, D. M. N. Prior, and N. J. Radcliff. "Our favourite compact graphs". Tech. Rep. 6, Edinburgh Concurrent Supercomputer Project, 1989.

[Ore 62] O. Ore. *Theory of Graphs*. Vol. 38, Amer. Math. Soc. Colloq. Publ., 1962. p. 35.

[Peas 77] M. C. Pease. "The undirected binary n-cube microprocessor array". *IEEE Transactions on Computers*, Vol. C-26, pp. 458–473, May 1977.

[Ples 81] J. Plesník. "The Complexity of Designing a Network with Minimum Diameter". *Networks*, Vol. 11, No. 1, pp. 77–85, 1981.

[Prad 82] D. K. Pradhan and S. M. Reddy. "A fault tolerant communication architecture for distributed systems". *IEEE Transactions on Computers*, Vol. C-31, pp. 863–870, September 1982.

[Prep 81] F. P. Preparata and J. V. Vuillemin. "The Cube Connected Cycles: A Versatile Network for Parallel Computers". *Communications of ACM*, Vol. 24, pp. 300–309, May 1981.

[Prio 90] D. M. Prior, M. Norman, N. Radcliffe, and L. Clarke. "What price Regularity?". *Concurrency: Practice and Experience*, Vol. 2, pp. 55–78, March 1990.

[Prov 83] J. Provan and M. Ball. "The complexity of counting cuts and of computing the probability that a graph is connected". *SIAM J. Comput.*, Vol. 12, No. 4, pp. 777–788, 1983.

[Raab 88] U. Raabe, M. Lobjinski, and M. Horn. "Verbindungsstrukturen für Multiprozessoren". *Informatik Spektrum*, No. 11, pp. 195–206, 1988.

[Ragh 86] C. S. Raghavendra and A. Varma. "Fault-Tolerant Multiprocessors with Redundant-Path Interconnection Networks". *IEEE Transactions on Computers*, Vol. C-35, pp. 307–316, April 1986.

[Rals 82] A. Ralston. "de Bruijn sequences. A model example of the interaction of discrete mathematics on the computer science". *Mathematics Magazine*, Vol. 55, No. 3, pp. 131–143, May 1982.

[Ramk 88] B. Ramkumar and L. V. Kale. "$\mathcal{D}$-Trees: A Class of Dense Regular Intercommection Topologies". In: *Frontiers '88, The 2nd Symposium on the Frontiers of Massively Parallel Computation*, pp. 207–210, IEEE, 1988.

[Reed 87a] D. A. Reed and R. M. Fujimoto. *Multicomputer Networks: Message-Based Parallel Processing. Scientific Computation Series*, MIT Press, 1987.

[Reed 87b] D. A. Reed and D. C. Grunwald. "The Performance of Multicomputer Interconnection Networks". *IEEE Computer*, pp. 63–73, June 1987.

[Rich 88] D. Richards and A. L. Liestman. "Generalizations of Broadcasting and Gossiping". *Networks*, Vol. 18, No. 1, pp. 125–138, 1988.

[Sabi 60] G. Sabidussi. "Graph Multiplication". *Math. Z.*, Vol. 72, pp. 446–457, 1960.

[Sain 94] C. F. Sainte-Marie. "Solution to problem number 58". *L'Intermédiaire des Mathématiciens*, No. 1, pp. 107–110, 1894.

[Sche 91] I. D. Scherson. "Orthogonal Graphs for the Construction of a Class of Interconnection Networks". *IEEE Transactions on Parallel and Distributed Processing*, Vol. 2, No. 1, pp. 3–19, January 1991.

[Seit 84] C. L. Seitz. "Concurrent VLSI Architectures". *IEEE Transactions on Computers*, Vol. C-33, No. 12, pp. 1247–1265, December 1984.

[Seit 85] C. L. Seitz. "The Cosmic Cube". *Communications of ACM*, Vol. 28, No. 1, pp. 22–33, January 1985.

[Seng 87] A. Sengupta, A. Sen, and S. Bandyopadhyay. "On an Optimally Fault-Tolerant Multiprocessor Network Architecture". *IEEE Transactions on Computers*, Vol. C-36, pp. 619–623, May 1987.

[Sieg 79] H. J. Siegel, R. J. McMillaen, and P. T. M. Jr. "A survey of interconnection methods for reconfigurable parallel processing system". In: *AFIPS Conference Proceedings*, pp. 529–542, NCC, June 1979.

[Skil 85] D. B. Skillicorn and R. Dawes. "Chordal Rings as Computer Networks". *Congr. Numer.*, No. 46, pp. 79–90, 1985.

[Skil 89] D. B. Skillicorn and B. Kocay. "A Global Measure of Network Connectivity". *Journal of Parallel and Distributed Computing*, Vol. 7, pp. 165–177, August 1989.

[Span 82] O. Spaniol. "Konzepte und Bewertungsmethoden für lokale Rechnernetze". *Informatik-Spektrum*, No. 5, pp. 152–170, 1982.

[Stor 70] R. M. Storwick. "Improved Construction Techniques for (d, k) Graphs". *IEEE Transactions on Computers*, Vol. C-19, pp. 1214–1216, December 1970.

[Sull 77] H. Sullivan and T. R. Bashkow. "A large scale, homogenous, fully distributed parallel machine". In: *Proceedings of the 4th Symposium on Computer Architecture*, pp. 105–124, 1977.

[Tain 76] M. Tainiter. "A new deterministic network reliability measure". *Networks*, No. 6, pp. 191–204, 1976.

[Tinh 80] G. Tinhofer. *Zufallsgraphen*. Vol. 17 of *Applied Computer Science*, Hauser, M"unchen,Wien, 1980.

[Toue 79] S. Toueg and K. Steiglitz. "The design of small-diameter networks by local search". *IEEE Transactions on Computers*, Vol. C-28, pp. 537–542, July 1979.

[Trel 87] P. C. Treleaven. *Computer architectures for artificial intelligence*, pp. 416–492. Vol. 272 of *Lecture Notes in Computer Science: Future Parallel Computer*, Springer Verlag, 1987.

[Trel 88] P. C. Treleaven. "Parallel architecture overview". *Parallel Computing*, No. 8, pp. 59–70, 1988. North Holland.

[Trip 85] S. K. Tripathi and S. T. Huang. "Circular shuffle network and distributed ressource scheduling". Tech. Rep. TR-1406, Dep. of Comp. Sci, University of Maryland, 1985. July.

[Upfa 84] E. Upfal. "Efficient Schemes for Parallel Communication". *Journal of the ACM*, Vol. 31, No. 3, pp. 507–517, July 1984.

[Vali 81] L. G. Valiant and G. J. Brebner. "Universal schemes for parallel communication". In: *Proc. 13th Annual ACM Symposium on Theory of Computing*, pp. 263–277, 1981.

[Weic 63] P. M. Weichsel. "The Kronecker Product of graphs". In: *Proceedings American Mathematical Society*, pp. 47–52, 1963. No. 13.

[Whit 32] H. Whitney. "Congruent graphs and the connectivity of graphs". *American Journal of Mathematics*, No. 54, pp. 150–168, 1932.

[Wilk 72] R. S. Wilkov. "Analysis and design of reliable computer networks". *IEEE Transactions on Communications*, Vol. CS-20, pp. 660–678, June 1972.

[Wils 79] R. J. Wilson and L. W. Beineke, Eds. *Applications of Graph theory*, Chap. Introduction, pp. 1–15. Academic Press, 1979.

[Witt 76] L. D. Wittie. "Efficient Message Routing in Mega-Micro-Computer Networks". In: *Proceedings of the 3rd Symposium on Computer Architecture*, pp. 136–140, 1976.

[Witt 81] L. D. Wittie. "Communication Structures for Large networks of Microcomputers". *IEEE Transactions on Computers*, Vol. C-30, pp. 264–273, April 1981.

[Wu 84] C. Wu and T. Feng, Eds. *Interconnection Networks for Parallel and Distributed Processing.* IEEE Computer Soc. NY, 1984.

[Yang 88] C. S. Yang, J. F. Wang, J. Y. Lee, and F. T. Boesch. "Graph theoretic reliability analysis for the Boolean n-cube network". *IEEE Transactions on Circuits and Systems*, Vol. CAS-35, pp. 1175–1179, September 1988.

[Yous 90] A. S. Youssef and B. Nahahari. "The Banyan-Hypercube Networks". *IEEE Transactions on Parallel and Distributed Systems*, Vol. 1, No. 2, pp. 160–169, April 1990.