BACKTRACK TO THE ORIGIN AND RETRY ROUTING FOR HYPERCYCLE-BASED INTERCONNECTION NETWORKS†

Nikitas J. Dimopoulos , Don Radvan, Kin F. Li
Electrical and Computer Engineering Department
University of Victoria
Victoria, B.C. Canada

ABSTRACT†

In this work, we present the Hypercycles, a class of multidimensional graphs, which are generalizations of the n-cube. These graphs are obtained by allowing each dimension to incorporate more than two elements and a cyclic interconnection strategy. Hypercycles, offer simple routing, and the ability, given a fixed degree, to chose among a number of alternative size graphs. These graphs can be used in the design of interconnection networks for distributed systems tailored specifically to the topology of a particular application. We are also presenting a back-track-to-the-origin-and-retry routing, whereupon paths that block at intermediate nodes are abandonned, and a new attempt is made. Intermediate nodes are chosen at random at each point from among the ones that form the shortest paths from a source to a destination. Preliminary simulation results are also presented.

1.0 Introduction

Message passing concurrent computers such as the Hypercube[11, 16], Cosmic Cube[15], MAX[12, 13], consist of several processing nodes that interact via messages exchanged over communication channels linking these nodes into one functional entity.

There are many ways of interconnecting the computational nodes, the Hypercube, Cosmic Cube, and the Connection Machine[17] having adopted a regular interconnection pattern corresponding to a binary n-dimensional cube, while MAX adopts a less structured, yet unspecified topology.

Several recent studies attempt extensions and generalizations of the basic tenets of the n-cube. Broder et. al. [4] have proposed product graphs[14] of small "basic" graphs. Their prime concern is to synthesize fault tolerant networks with a given degree of coverage. In these multidimensional graphs, they define a single route from a source to a destination, as the product of routes in each of the constituent dimensions. Routing is exhausted in each dimension before another dimension is considered. Bhunyan and Agrawal [3] have introduced the generalized hypercubes (GHC) which are also graph products of fully connected "basic" graphs. The mixed radix system [2] is used to express the properties of these graphs and their routing. Wittie [18] gives a good overview and comparison of several interconnection networks including the spanning bus and dual bus hypercubes. These are essentially binary n-cubes with broadcast busses connecting the processors in each dimension.

The advantages of having a regularly structured interconnection are many-fold, and they have been proven time and again in their being incorporated in many recent designs [6,11,12,13,15,16,17]. In these structures, easy deadlock-free routing [7] can be accomplished by locally computing each successive intermediate node -for a path that originates at a source node and terminates at a destination node- as a function of the current position and the desired destination. Many regular problems (such as the ones found in image processing, physics etc.) have been mapped on such regular structures, and run on the corresponding machines exhibiting significant speedups. In contrast, embedded real-time applications, such as the ones addressed by the MAX project [12, 13], tend to exhibit variable structures that do not necessarily map optimally to an n-cube. In addition, since the size of a binary n-cube is given as 2^n (n being the degree of the graph), it means that a particular configuration cannot be expanded but in predefined quantum steps. For example, if a given embedded application requires a system comprised of 9 nodes, the next larger n-cube with 16

nodes must be chosen. This constitutes a significant increase in resource allocation, especially in the light of the power-mass limited environment of a spacecraft.

Hypercycles[9] can be considered as products of "basic" graphs that allow, as compared to the Generalized Hypercubes (GHC) [3], a richer set of component "basic" graphs ranging in complexity from the simple rings to the fully connected ones used in the GHC. Also, contrary to Broder et. al.[4], we define the component graphs and provide analytical expressions for routing, our aim being twofold:

(a) To provide computer interconnection networks that match the node requirements of a given embedded system. Since our primary target is spacecraft applications which are weight and power limited, (the nodes in a spacecraft computer network are the primary weight contributor rather than the communication media) the exact matching of the node requirements is of paramount importance.

(b) To increase throughput of a given network by providing routing expressions that can be computed analytically (and hence are candidates for VLSI implementation) and which provide a maximum number of alternate paths from a source to a destination. The existence of alternate paths guarantees that a message will not be blocked waiting for its single route to be freed, but it would in turn search for the availability of alternate paths. This strategy also provides for fault protection, since a faulty path can be marked permanently busy, and thus messages can be routed around it. (Such an approach of adaptive routing is applied in the hypercube through the Hyperswitch [6])

The Hypercycles, being regular graphs, retain the advantages of easy routing and regularity. Yet, since we are dealing with a class, rather than isolated graphs, we have the flexibility of adopting any particular graph (from the class) that closely matches the requirements of a given application. Since the graphs belong to the same class, routing is accomplished via the same methods and thus the same hardware can conceivably be used to configure structures with different sizes and topologies.

This work describes such a class of generalized interconnection networks, with routing strategies that are similar to that of the n-cube [4]. Yet, these networks offer richer topologies, and contain both the n-cube and the ring as special cases. While the n-cube is based on representation of nodes in base 2, we generalize by using the mixed radix system representation. Such a representation, includes the binary (and hence the n-cube) as well as the arbitrary base b representation as special cases.

This work is divided into three parts. Section 2.0 introduces the Mixed Radix System, Section 3.0 presents some basic graph terminology and notation, while Section 4.0 introduces the Hypercycles and discusses their properties.

2.0 Mixed Radix Number System

The mixed radix representation [2], is a positional number representation, and it is a generalization of the the standard b-base representation, in that it allows each position to follow its own base independently of the other.

Thus, given a decimal number M factored into r factors m_1, m_2, m_3 ... $, m_r$ as $M = m_1 \times m_2 \times m_3 \times \cdots \times m_r$ then any number $0 \leq X \leq M-1$ can be represented as the following r-tuple

$$(X)_{m_1 m_2 \ldots m_r} = x_1 x_2 \ldots x_r |_{m_1 m_2 \ldots m_r}$$

where $0 \leq x_i \leq (m_i - 1)$; $i = 1,2,\ldots,r$ and the x_i's are chosen in such a way so as $X = \sum_{i=1}^{r} x_i w_i$ where $w_i = \dfrac{M}{m_1 m_2 \cdots m_i}$

We use the notation $(X)_{m_1 m_2 \ldots m_r} = x_1 x_2 \ldots x_r |_{m_1 m_2 \ldots m_r}$ to indicate the radices involved. Since for most cases we shall be dealing with a single set of radices, $m_1, m_2, \ldots, m_r$, we shall omit, when obvious, the

radix indication from the notation $x_1x_2...x_r |_{m_1 m_2 ...m_r}$.

As an example, if we chose $M = 10 = 2 \times 5$, then any number between 0 and 9 can be represented with two digits, the first one ranging from 0 to 1, and the second one from 0 to 4. Thus $(6)_{2,5} = 11|_{2,5}$, since $m_1 = 2$, $m_2 = 5$, $w_1 = M / m_1 = 10/2 = 5$, and $w_2 = M/ (m_1 m_2) = 1$. Therefore,

$(6)_{2,5} = 11|_{2,5} = 1 \times w_1 + 1 \times w_2 = 1 \times 5 + 1 \times 1 = 6|_{10}$.

Similarly, we can see that the mixed radix system representation, is a generalization of the standard base-b system. Indeed, if we select $m_1 = m_2 = ... = m_r = b$, then $M = b^r$, the corresponding weights w_i become $w_i = M/b_i = b^{r-i}$, therefore, the representation of a number X in

base $b \times b \times ... \times b$ becomes $(X)_{bb...b} = \sum_{i=1}^{r} x_i w_i = \sum_{i=1}^{r} x_i b^{r-i}$

which is exactly the representation in base-b.

3.0 Graph Notation

An undirected graph G is defined as the following tuple: $G = (N, E)$, where N is the set of nodes (vertices) $N = \{ \alpha_i ; i = 1,2,...N \}$, and E the set of edges defined as

$E = \left\{ e_{ij_i} = \left(\alpha_i, \beta_{j_i} \right) j_i = 1,2,...,d_i ; i = 1,2,...,N \right\}$

with $\alpha_i, \beta_{j_i} \in N$ and where d_i is the degree of node α_i (i.e. the number of edges incident at a particular node). The degree of a graph, denoted d(G), is defined as the maximum of the node degrees. A walk in G [5] is a sequence of edges $e_1 e_2 ... e_l$, such that if $e_i = (\alpha_i, \alpha_{i+1})$ then $e_{i+1} = (\alpha_{i+1}, \alpha_{i+2})$ and $e_i \in E$. The length l of such a walk is defined as the number of edges occurring in it. The distance, $dis(\gamma, \delta)$, between nodes γ and δ is defined as the shortest walk between γ and δ if any, otherwise, $dis(\gamma, \delta) = \infty$. The diameter of a graph, denoted by k, is defined as the maximum distance between any pair of nodes. A graph is regular, if all nodes have the same degree.

4.0 Hypercycles.

We define now an r- dimensional Hypercycle, as the following regular undirected graph:

$$G_m^\rho = \left\{ N_m^\rho, E_m^\rho \right\}$$

where $m = m_1,m_2,m_3,...,m_r$ a mixed radix, $\rho = \rho_1,\rho_2,...,\rho_r$; $\rho_i \leq m_i / 2$ the connectivity vector, determining the connectivity in each dimension which ranges from a cycle ($\rho_i = 1$) to the fully connected ($\rho_i = \lfloor m_i/2 \rfloor$), and $N_m^\rho = \{0,1,2,...,M-1\}$. Each node $\alpha \in N_m^\rho$ is represented in the mixed radix system $m_1,m_2,m_3,...,m_r$ as

$(\alpha)_{m_1 m_2 ...m_r} = \alpha_1 \alpha_2...\alpha_r$. The set of edges $(N_m^\rho)^2 \supseteq E_m^\rho$ is

defined in the following way. Given $\alpha, \beta \in N_m^\rho$ and if α, β are

represented as $(\alpha)_{m_1 m_2 ...m_r} = \alpha_1 \alpha_2 ... \alpha_r$ and

$(\beta)_{m_1 m_2 ...m_r} = \beta_1 \beta_2...\beta_r$. Then $(\alpha, \beta) \in E_m^\rho$ if and only if there

exists $1 \leq j \leq r$ such that

$\beta_j = (\alpha_j \pm \xi_j) \bmod m_j$ with $1 \leq \xi_j \leq \rho_j$
and $\alpha_i = \beta_i ; i \neq j$

Observe that since $\beta_j = (\alpha_j \pm \beta_j) \bmod m_j$, E_m^ρ is symmetric.

Example: Suppose that $M = 12 = 4 \times 3 = m_1 \times m_2$. and

$\rho = 1,1 = \rho_1,\rho_2$. Therefore, for the graph generated by using the above mixed radix and connectivity vector, node $(1)_{4,3} = 01 = \alpha_1 \alpha_2$, is distance one away from node $(10)_{4,3} = 31 = \beta_1 \beta_2$. Indeed, according to our definition, $\beta_1 = 3 = (0 - 1) \bmod 4 = (\alpha_1 - \xi_1) \bmod m_1$, and $\alpha_2 = \beta_2 = 1$.

The previously defined Hypercycles, are regular graphs of degree[§]

$$d = \sum_{i=1}^{r} f(m_i, \rho_i) \quad \text{where}$$

$$f(m_i, \rho_i) = \begin{cases} 2\rho_i & \text{if} \quad 2\rho_i < m_i \\ m_i - 1 & \text{if} \quad 2\rho_i = m_i \end{cases}$$

and diameter k [1][†]. $k = \sum_{i=1}^{r} \left\lceil \dfrac{\lfloor m_i /2 \rfloor}{\rho_i} \right\rceil$

It is easy to see that the n-cube is a Hypercycle, obtained with $M = 2 \times 2 \times \cdots \times 2 = 2^n$ and $\rho = 1,1,1,...,1$.

4.1 Routing

Hypercycles, have routing properties that are similar to those of the n-cube. From the definition of the Hypercycle, nodes

$(\alpha)_{m_1 m_2 ...m_i ...m_r} = \alpha_1 \alpha_2....\xi...\alpha_r$ and

$(\alpha^*)_{m_1 m_2 ...m_i ...m_r} = \alpha_1 \alpha_2....\xi...\alpha_r$

are at most distance $\left\lceil \dfrac{\lfloor m_i /2 \rfloor}{\rho_i} \right\rceil$ away. A walk, from node α to node

α^*, of length

$l_{max} = \left\lceil \dfrac{\lfloor m_i /2 \rfloor}{\rho_i} \right\rceil$ can be constructed as follows:

$\alpha_1 \alpha_2...\alpha_i...\alpha_r$, $\alpha_1 \alpha_2...\xi_1...\alpha_r$, $\alpha_1 \alpha_2...\xi_2....\alpha_r$, ...,

$\alpha_1 \alpha_2...\xi...\alpha_r$.

such that

$$\xi_{j_i +1} = \begin{cases} \xi_{j_i} + \rho_i & \text{if} \quad \left[(\xi - \xi_{j_i}) \bmod m_i = |\xi_{j_i}, \xi| \right] > \rho_i \\ \xi_{j_i} - \rho_i & \text{if} \quad \left[(\xi_{j_i} - \xi) \bmod m_i = |\xi_{j_i}, \xi| \right] > \rho_i \\ \xi & \text{if} \quad |\xi_{j_i}, \xi| \leq \rho_i \end{cases}$$

$\xi_0 = \alpha_i \qquad \xi_{l_{max}} = \xi$

We call the length l_{max} of such a walk, the distance along dimension i

---

Given an origin $(\alpha)_{m_1 m_2 ... m_r} = \alpha_1 \alpha_2 ... \alpha_r$

and a destination $(\beta)_{m_1 m_2 ... m_r} = \beta_1 \beta_2 ... \beta_r$

and if q_i denotes their distance along dimension i, their total distance, denoted as $dis(\alpha,\beta)$ and defined as the sum of the individual distances along all the dimensions, is given as

$$dis(\alpha, \beta) = q = \sum_{i=1}^{r} q_i$$

For these nodes, there are a total of [‡]

$$I = \binom{q}{q_1, q_2, ..., q_r} = \frac{q!}{q_1! \, q_2! \cdots q_r!}$$

distinct walks of length q that connect them . These paths can be constructed by sequentially modifying the source address, each time substituting a source digit by an intermediate walk digit , as specified in equation (2) above, until the destination is reached. As an example, the following walk connects the source to the destination.

$$\text{source} = \alpha_1 \alpha_2 \alpha_3 ... \alpha_r \, , \, \alpha_1 \xi_1 \alpha_3 ... \alpha_r \, , \, \alpha_1 \xi_1 \psi_1 ... \alpha_r$$

$$\alpha_1 \xi_2 \psi_1 ... \alpha_r \, , \, \alpha_1 \xi_2 \psi_2 ... \alpha_r \, , \, ..., \, \alpha_1 \xi_2 \beta_3 ... \alpha_r \, , \, ..., \, \beta_1 \beta_2 \beta_3$$

$$... \beta_r = \text{destination}$$

Figure 1., gives an example of two distinct walks of equal length that connect a source to a destination, for a Hypercycle.

4.2 Average Distance Calculation

Given an r-dimensional generalized graph G_m^ρ , and restricting ourselves in any single dimension i, we can calculate the number of nodes that are distance l away from a source node $(\alpha)_{m_1 m_2 ... m_r} = \alpha_1 \alpha_2 ... \alpha_r$ as equal to the number of symbols ξ_i that satisfy the relation $(l-1)\rho_i < |\alpha_i, \xi_i| \le l\rho_i$. This number is given as

$$n_l^i = \begin{cases} 2\rho_i & \text{if} \quad l\rho_i < \lceil m_i / 2 \rceil \\ (m_i-1)-2(l-1)\rho_i & \text{if} \quad \lceil m_i/2 \rceil \le l\rho_i \le \frac{m_i-1-\rho_i}{2} \\ 0 & \text{otherwise} \end{cases}$$

Given a node $(\alpha)_{m_1 m_2 ... m_r} = \alpha_1 \alpha_2 ... \alpha_r$ there exist

$$n_1 = \sum_{i=1}^{r} n_1^i \quad \text{nodes at distance one, and in general}$$

$$n_l = \sum_{\substack{l_1, l_2, ..., l_r \ge 0 \\ l_1 + l_2 + ... + l_r = l}} \prod_{\substack{i=1 \\ l_i \ne 0}}^{r} n_{l_i}^i$$

nodes at distance l .This equation is simplified for the case where $\rho_i = \lfloor m_i / 2 \rfloor$ to the following:

$$n_l = \sum_{i_1=1}^{r-l+1} \sum_{i_2=i_1+1}^{r-l+2} \cdots \sum_{i_l=i_{l-1}+1}^{r} (m_{i_1}-1)(m_{i_2}-1)\cdots(m_{i_l}-1)$$

Therefore, the average distance between any two nodes in a Hypercycle G_m^ρ can be calculated as $\overline{d} = \dfrac{\sum_{l=1}^{r} l n_l}{m_1 m_2 \cdots m_r}$

Some typical distances are given in TABLE 1.

[‡] For the definition of a multinomial number, see [1] pp 32.

5.0 Deadlock Avoidance in Routing.

In section 4.1, we have proven that the Hypercycles are connected graphs, and we have given a method that establishes at least one path from a source to a destination node. In this part, we are concerned with optimally chosing one of the paths. Routing must be efficient and deadlock free. Deadlock occurs when resources (in this case node to node communication segments) are allocated so that the completion of a partial path requires a segment already allocated to a different partial path which in turn waits for a segment in the first partial path. It is obvious that no messages can propagate over the deadlocked paths, and the only remedy is to break the already established and deadlocked partial paths and try again.

Deadlock may occur easily in cases where the segments that form the paths are chosen at random. Certain routing algorithms (e.g. virtual channels, e-cube routing[7]) avoid deadlocks by ordering the resources (channels) to be allocated. Thus a lower order resource cannot be committed if a needed higher order resource cannot be obtained. The disadvantage of this approach in an interconnection network is that it limits the number of paths connecting a source to a destination to exactly one, even though several alternate free paths may exist at a particular moment. On the other hand, discovering a deadlock and breaking it may prove expensive. Nevertheless, for certain operating conditions, it may prove beneficial to adopt a strategy where deadlocks are broken by requiring a blocked path to backtrack to its origin and retry.

6.0 Backtrack-to-the-origin-and-retry routing

For Hypercycle-based interconnection networks, because of the existence of cycles in each dimension, the use of an e-cube type routing that avoids deadlocks, is impractical. We are proposing a deadlock detecting routing strategy. According to our backtrack-to-the-origin-and-retry routing, we identify at each node all potential candidates for next node in the path according to equation 4.1.1. For all such identified nodes, we also identify the corresponding free ports that can be used in order to continue the path. One of these free ports is randomly selected through which the subsequent link in the path is established. If no free ports are to be found at an intermediate node in the path, then a break is returned to the origin (through the already established partial path to the blocking node), the partial path is destroyed, and a new attempt for the creation of the required path is initiated. This routing strategy avoids deadlocks through the backtracking, and also guarantees that the formed path will be of a minimum length, since each subsequent link is selected according to equation 4.1.1.

We have used Extend™[†] to construct a simulator capable of simulating any Hypercycle based network. For this simulator, we implemented both the backtrack-to-the-origin-and-retry as well as the e-cube routing strategies. The e-cube routing can only be used for binary cube networks. For each node, we assumed a Poisson message generator which generates packets with uniform distribution of destinations. Each packet carries both the destination addresse which is used for routing. Links are assigned priorities, so that collisions can be resolved. We assumed a packet transmission time (over an established source to destination path) of 100 simulation-clock ticks. We use the simulator to obtain the throughput and delay characteristics of several networks for both e-cube and backtrack-to-the-origin-and-retry in terms to the offered load. Both the offered load and the throughput were normalized in terms to the maximum capacity of each network taken to be proportional to the number of links in the corresponding graph. The average delay was expressed in actual time units necessary to establish a source-to-destination circuit. Preliminary simulation results are depicted in figs. 2 and 3.

As it was expected, the performance of the backtrack-to-the-origin-and-retry for both binary cubes and hypercycles of similar sizes, is clearly superior that of the e-cube as it can be seen in figs. 2 and 3. This is attributed to the fact that the backtrack-to-the-origin-and-retry can use alternative paths to the destination instead of the single path alloted by the e-cube routing. The additional advantage of the backtrack-to-the-origin-and-retry is its inherent fault tolerance. Indeed, if one of links in the

---

[†] Extend is a trademark of Imagine That inc.

TABLE 1.

GRAPH			DEGREE	DIAMETER	NODES	AVERAGE DISTANCE PER NODE
m	=	6 2	6	2	12	1.454
ρ	=	3 1				
m	=	5 3	6	2	15	1.571
ρ	=	2 1				
m	=	5 2 2	6	3	20	1.894
ρ	=	2 1 1				
m	=	3 3 3	6	3	27	2.077
ρ	=	1 1 1				
m	=	3 3 2 2	6	4	36	2.4
r	=	1 1 1 1				
m	=	3 2 2 2 2	6	5	48	2.723
ρ	=	1 1 1 1 1				
m	=	2 2 2 2 2 2	6	6	64	3.047
ρ	=	1 1 1 1 1 1				
m	=	2 2 17	6	6	68	3.403
ρ	=	1 1 2				
m	=	2 2 3 7	6	6	84	3.434
ρ	=	1 1 1 1				
m	=	3 5 7	6	6	105	3.615
ρ	=	1 1 1				
m	=	5 5 5	6	6	125	3.629
ρ	=	1 1 1				
m	=	2 2 2 2 2 2 2	7	7	128	3.528
ρ	=	1 1 1 1 1 1 1				

Figure 2. Throughput vs. offered load using backtrack to the origin and e-cube routing. The offered load and throughput are normalized to the capacity of the interconnection network.

Figure 3. Delay vs. offered load using backtrack-to-the-origin and e-cube routing. The offered load is normalized to the capacity of the interconnection network.

a. Hypercycle $\mathcal{G}^{11}_{43}$

b. Binary 3-cube $\mathcal{G}^{111}_{222}$
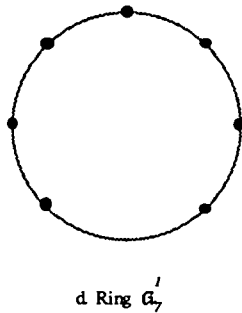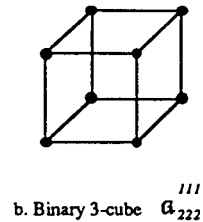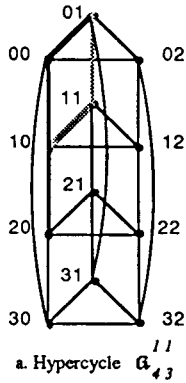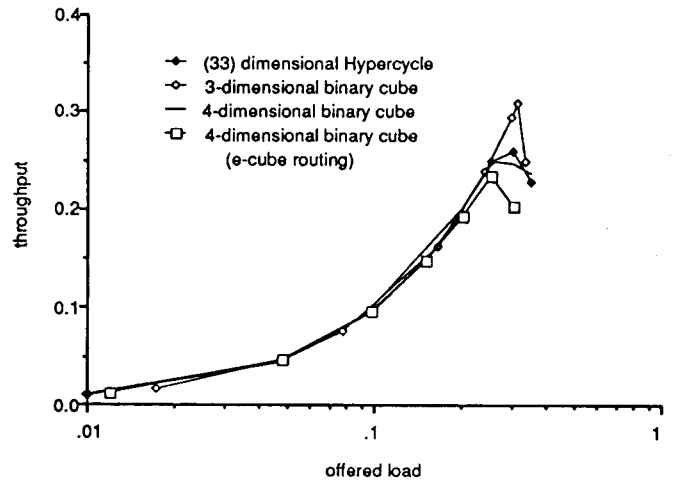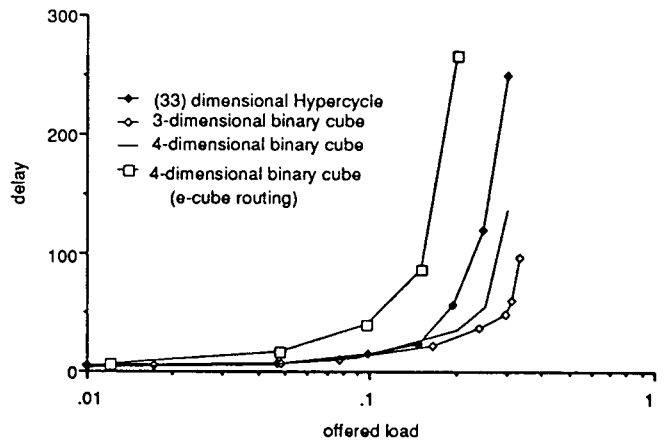
c. Illiac IV $\mathcal{G}^{11}_{88}$

d. Ring $\mathcal{G}^{1}_{7}$

Figure 1. Examples of Hypercycles.

network failed, it could be marked as permanently busy, and packets would be routed around it. This obviously is not the case for the e-cube routing.

6.0 Conclusions and Discussion

In this work, we presented the Hypercycle, a class of multidimensional graphs, which are essentially generalizations of the n-cube. These graphs are obtained by allowing each dimension to incorporate more than two elements and a cyclic interconnection strategy.

Although these graphs are not the densest possible, they are attractive, because of their simple routing. Similarly to the n-cube, the destination address is used to sequentially route a message through intermediate nodes as outlined in section 4.1. Also, since the node addresses are represented in a mixed radix as a sequence of r-digits, each one of these digits is processed independently and in parallel with the remaining digits. Thus the hardware involved in the routing can be made fast (because of the parallelism) and simple (since each module need only handle arithmetic modm_i, as compared to arithmetic mod$m_1 m_2...m_r$ needed when all the address digits are necessary as is the case with such networks as the chordal rings [10], or the cube connected cycles [5]).

For these graphs, we have also calculated the average distance between any two nodes. Assuming that the graph is the representation of a multiprocessor, the average distance provides a metric of the communication complexity involved in such a structure. Compare for example the average distances of the 7-cube and the ($m = 5\ 5\ 5$, $\rho = 1\ 1\ 1$) graph. The average distances, as they can be determined from TABLE 1, are 3.528 and 3.629 respectively, both have similar number of nodes, while their diameters are equal to six and seven respectively.

The graphs presented in this study, are generalizations of some well known graphs such as the binary n-cube, 2- and 3-dimensional meshes, and rings, which are included as special cases. Examples of some special cases are depicted in Figure 1.

The Hypercycles, can be further extended by employing a different connectivity strategy in each dimension. Thus, in its most general form, eqn. 1. can be modified into

$$\beta_j = (\alpha_j + \xi_j) \text{mod} m_j \ ; \ \xi_j \in \left\{ \xi_j^1, \xi_j^2, ..., \xi_j^k \right\}$$

provided that the subgraph (projected in dimension j) is connected. Examples of such extensions include the chordal rings [10], cube connected cycles [5] etc. The disadvantage is the increased complexity involved in the routing calculations for that particular dimension.

In conclusion, Hypercycles are generalizations of the basic n-cube and they offer simple routing, and the ability, given a fixed degree, to chose among a number of alternative size graphs. The use of the backtrack-to-origin-and-retry routing strategy offers incresed throughput and fault tolerance. These properties are important for the design of concurrent systems (e.g. MAX [12], Hypercube [11]) of varying size and connectivity, and tailored specifically to the topology of a particular application.

REFERENCES

1. Berge C., *Principles of Combinatorics* Academic Press, New York and London, 1971.
2. Bhuyan, L. N., and D. P. Agrawal, "Design and Performance of Generalized Interconnection Networks" *IEEE Trans. Comput.* Vol. C-32, No. 12, pp. 1081-1090, Dec. 1983.
3. Bhuyan, L. N., and D. P. Agrawal, "Generalized Hypercube and Hyperbus Structures for a Computer Network" *IEEE Trans. Computrs.* Vol. C-33, No. 4. pp. 323-333, (Apr. 1984).
4. A. Broder, D. Dolev, M. Fischer, B. Simons "Efficient Fault Tolerant Routings in Networks" *Proceedings of the 16th Annual ACM Symposium on the Theory of Computing* pp. 536-541, (May 1984).
5. Carlsson, G. E. , J. E. Cruthirds, H. B. Sexton, and C. G. Wright "Interconnection Networks Based on a Generalization of Cube-Connected Cycles" *IEEE Trans. Comput.*, Vol. C-34, No. 8, pp. 769-772, Aug. 1985.
6. E. Chow, H. Madan, J. Peterson "A Real-Time Adaptive Mesage Routing Network for the Hypercube Computer" *Proceedings of the Real-Time Systems Symposium*, pp. 88-96, San Jose CA., (Dec. 1987)
7. Dally, W.J., and C. L. Seitz "Deadlock-Free Message Routing in Multiprocessor Interconnection Networks" *IEEE Trans. Comput.* Vol. C-36, No. 5, pp. 547-553, May 1987.
8. C. Duret " A Nodal Structure Switching Network" *Proceedings Colloque International de Communication* Vol. 2, pp. 834-839, Paris, France (May 1979).
9. Feng, T. Y. ,"A Survey of interconnection Networks" *IEEE Computer*, Vol. 14, pp. 12-27, Dec. 1981
10. Imase, M., T. Soneoka, and K. Okada, "Connectivity of Regular Directed Graphs with Small Diameters" *IEEE Trans. Comput.*, Vol. C-34, No. 3, pp. 267-273, Mar. 1985.
11. Peterson, J.C., J. O. Tuazon, D. Lieberman, M. Pniel "The MARK III Hypercube -Ensemble Concurrent Computer" *Proceedings of the 1985 International Conference on Parallel Processing* pp. 71-73, Aug. 20-23 1985.
12. Rasmussen, R. D., G. S. Bolotin, N. J. Dimopoulos, B. F. Lewis, and R. M. Manning "Advanced General Purpose Multicomputer for Space Applications" *Proceedings of the 1987 International Conference on Parallel Processing*. pp. 54-57. (Aug. 1987)
13. Rasmussen, R. D., N. J. Dimopoulos, G. S. Bolotin, B. F. Lewis, and R. M. Manning "MAX: Advanced General Purpose Real-Time Multicomputer for Space Applications" *Proceedings of the IEEE Real Time Systems Symposium* pp. 70-78, San Jose, CA.,(Dec. 1987).
14. G. Sabidussi "Graph Multiplication" *Math. Zeitschr.* Vol. 72, pp. 446-457 (1960).
15. Seitz, C. L. "The cosmic cube" *CACM* vol. 28, pp.22-33, Jan. 1985
16. Tuazon, J. O., J. C. Peterson, M. Pniel, and D. Lieberman "Caltech/JPL MARK II Hypercube Concurrent Processor"*Proceedings of the 1985 International Conference on Parallel Processing* pp. 666-673, Aug. 20-23 1985.
17. Waltz, D. L. "Applications of the Connection Machine" *IEEE Computer* January 1987, pp.85-97
18. L. D. Wittie "Communication Structures for Large Networks of Microcomputers" *IEEE Trans. on Computers* Vol. C-30, No. 4, pp.264-273. (Apr. 1981).