The Hierarchical Cliques Interconnection Network

Stuart Campbell† and Mohan Kumar† and Stefan Olariu†

†Department of Computer Science, Curtin University of Technology, GPO Box U1987, Perth, W.A. 6001, Australia, and ‡Department of Computer Sciences, College of Sciences, Old Dominion University, Norfolk, Virginia 23529-8162.
E-mail: {campbell, kumar}@cs.curtin.edu.au

The fully connected network possesses extremely good topological, fault-tolerant, and embedding properties. However, due to its high degree, the fully connected network has not been an attractive candidate for building parallel computers. On the other hand, tree-based networks are popular as parallel computer networks, even though they suffer from poor fault-tolerance and embedding properties. The hierarchical cliques interconnection network described in this paper incorporates the positive features of the fully connected network and the tree network. In other words, the hierarchical cliques possess such desirable properties as low diameter, low degree, self routing, versatile embedding, good fault-tolerance and strong resilience. Hierarchical cliques can embed most important networks and possess a scalable, modular structure. High local connectivity of the topology combined with efficient routing schemes facilitate efficient execution of algorithms exhibiting dense local communication. Complexity analysis of Floyd’s all-pairs shortest path algorithm demonstrates that good performance can be obtained by a widely useful algorithm on hierarchical cliques.

Key Words: cliques, trees, interconnection networks, network embedding, fault-tolerance

DRAFT January 12, 2000, 10:33am DRAFT
1. INTRODUCTION

In recent years several interconnection networks have been proposed and studied for their suitability in parallel computers. In general, some of the desirable properties of multiprocessor topologies include low diameter, low degree, high bisection width, versatile embedding properties and robustness to faults. The fully connected network possesses all the desirable properties except low degree. As the degree of a fully connected network of size \( N \) is \( (N - 1) \), it is very expensive to construct multiprocessor networks with this topology. Less costly topologies, such as meshes and \( k \)-ary \( n \)-cubes, have been the most popular for building commercial parallel computers. Mesh topologies map well onto a plane, and are well suited to VLSI implementation, but suffer from high diameter and correspondingly high latency. Methods such as wormhole routing [4] have been used in mesh architectures to reduce latency. These methods are very successful when contention is low, but are less effective when network congestion rises [1]. Under these circumstances meshes are only effective if the algorithm exhibits communication locality. Hence they are suitable for implementation of applications such as computational fluid dynamics and low level image processing. Other topologies which allow better use to be made of communication locality offer greater performance under these circumstances. Meshes also suffer from poor embedding properties and fault-tolerance. The binary \( n \)-cube topology possesses such desirable properties as symmetry, regularity, robustness, logarithmic diameter and self routing schemes, among many others. On the other hand the hypercube topology has high degree for large networks and it does not have a modular structure. The tree network has low degree and is suitable for VLSI implementation. However, the tree network is notorious for message congestion at the root. On-going research is directed towards developing new interconnection networks that combine the good features of the hypercube, the mesh, the tree, and the fully connected network [9, 16, 5, 8, 12, 10]. The fat-tree has been popular, its topology was adopted in the Connection Machine - 5 (CM-5),
a commercial machine. Further, Leiserson [9] and later Öhring et al. [13] have demonstrated the versatility of the fat tree network. Unlike the tree network, the intermediate nodes of the fat tree network have multiple parent nodes and hence there are multiple communication paths among all pairs of pendant nodes.

In this paper we describe the hierarchical cliques (HiC) network, which combines positive features of the fully connected network, the extended hypercube network, and the tree network. The HiC is a k-ary tree, modified to enhance local connectivity in a hierarchical, modular fashion. The HiC structure can be used to define a number of different useful multiprocessor architectures. We define only those nodes on level 0, the leaf nodes, as processor elements (PEs). All other nodes are switching elements (SEs). The cost effectiveness of the HiC has already been analysed in [2], where it was referred to as the Reliable Hierarchical Cliques (RHHiC). At the risk of causing some confusion with a similar, but inferior topology referred to in [2] as HiC, we have changed the name to prevent the misapprehension that reliability is the main feature of this topology. Here we analyse some of its topological features relevant to performance and fault tolerance. We demonstrate the simplicity of various routing algorithms using the proposed addressing scheme and define some embeddings of popular topologies onto the HiC to demonstrate its versatility. Finally, we evaluate the communication performance of the HiC using simulation.

The paper is organised as follows. Section 2 describes the structure and addressing scheme of the HiC. Some topological parameters of the HiC are described in Section 3. Section 4 demonstrates the fault tolerant properties of the HiC. Various message routing algorithms for the HiC are described in Section 5. Embeddings of some important topologies onto the HiC are discussed in Section 6. Simulation results in Section 7 are used to demonstrate the performance of the HiC. Section 8 concludes the paper.
2. STRUCTURE OF THE HIERARCHICAL CLIQUES

$HiC_{(k,h)}$ is a $k$-ary tree of height $h$ modified so that groups of nodes on the same level form cliques. Members of a clique are referred to as *neighbours*. The root node is at level $h$, and has address 0. The $k$ children of the root node are at level $h-1$ and form a clique. They have addresses consisting of a single unique digit in the range 1 to $k$. In general let $\mu$ be a node at level $l$ of $HiC_{(k,h)}$, where $(0 \leq l < h - 1)$. Then $\mu$ has address $M$ consisting of a sequence of digits $\langle M_1, \ldots, M_{h-1} \rangle$, where each digit is in the range 1 to $k$. Consider a second node $\nu$ in the same $HiC$ as $\mu$. If $\nu$’s address $N$ is a proper suffix of $M$ then $\nu$ is an ancestor of $\mu$ and $\mu$ is a descendant of $\nu$. If $M = \langle M_1, N \rangle$ then $\nu$ is the parent of $\mu$ and $\mu$ is a child of $\nu$. If a sequence $P$ exists such that $N = \langle P, N_{h-1} \rangle$ and $M = \langle P, M_{h-1} \rangle$ then $\mu$ and $\nu$ are neighbours.

We refer the reader to Figure 1, illustrating the structure and addressing scheme of an $HiC$ with $k = 4$ and $h = 3$. (Only one quarter of the level 0 nodes are shown). The root node has address 0 and is at level 3. The root node has four children at level 2, with addresses 1, 2, 3 and 4. The nodes at level 2 form a clique; in Figure 1 nodes of a clique are shown enclosed in a dashed oval. Each level 2 node has four children at level 1. The address of a node at level 1 consists of the address of its parent node appended to a digit between 1 and 4, which distinguishes the level 1 node from its siblings. Thus, nodes 12, 22, 32 and 42 are all children of node 2.

Nodes at level 1 are neighbours if their parents are neighbours and the first digit of their address is the same; for example nodes 21, 22, 23 and 24 form a clique. The address of a node at level 0 consists of the address of its parent node appended to a digit between 1 and 4. The digit distinguishes the level 0 node from its siblings. Thus nodes 141, 241, 341 and 441 are all children of node 41. Nodes at level 0 are neighbours if their parents are neighbours and the first digit of their address is the same, for example nodes 241, 242, 243 and 244 form a clique.
Represents a single connection between two nodes.

Represents connections forming a clique among connected nodes.

**FIG. 1.** Part of a $HiC$ with $k = 4$ and $h = 3$. 
3. PARAMETERS OF THE HIC

Assume that \( k \neq 1 \), then the nodes of \( HiC_{(k,h)} \) occur at \( h + 1 \) levels, with the leaves at level 0 and the root at level \( h \). The total number of nodes in \( HiC_{(k,h)} \) is \( N = \frac{k^{h+1} - 1}{k-1} \). Recall that we use only leaf nodes as processors (PES), with all other nodes acting as switches (SEs). We wish to determine the number of SEs \( (N_S) \), the number of PEs \( (N_P) \) and the number of communication links \( (L) \). An easy inductive argument shows that at level \( l \), \( (0 \leq l \leq h) \), there are \( N_l = k^{h-l} \) nodes. Therefore, the number of leaf nodes is \( N_P = k^h \). The number of switching nodes \( N_S = \sum_{l=1}^{h} k^{h-l} = \frac{k^{h+1} - 1}{k-1} \). The number of links \( L = L_T + L_C \) where \( L_T \) represents the number of links in a \( k \)-ary tree of height \( h \) and \( L_C \) represents the number of links connecting nodes into \( k \)-cliques. The total number of links within a \( k \)-ary tree of height \( h \) is given by \( L_T = \sum_{l=1}^{h} k^l \). The number of links within a \( k \)-clique is given by \( k(k-1)/2 \), and the number of \( k \)-cliques within \( HiC_{(k,h)} \) is given by \( \sum_{l=0}^{h-1} k^{h-1-l} \) so \( L_C = (k-1)/2 \times \sum_{l=1}^{h} k^l \). Therefore \( L = \frac{k(k+1)(k^h-1)}{2(k-1)} \).

One of the most important properties of a multiprocessor interconnection network is the diameter. Before deriving the diameter of an \( HiC_{(k,h)} \) we require two lemmas.

**Lemma 3.1.** In any \( HiC_{(k,h)} \), if node \( \mu \) has address \( M = \langle M_0, \ldots, M_{h-1} \rangle \) and node \( \nu \) has address \( N = \langle N_0, \ldots, N_{h-1} \rangle \) such that \( M_x \neq N_x \) for \( 0 \leq x \leq h-1 \), then \( d(\mu, \nu) = 2h-1 \).

**Proof.** Since \( M_x \neq N_x \) for \( 0 \leq x \leq h-1 \) no prefix \( P \) can exist such that \( M = \langle P, M_{h-1} \rangle \) and \( N = \langle P, N_{h-1} \rangle \). Therefore \( \mu \) and \( \nu \) cannot be neighbours, nor can their ancestors at level \( l < h-1 \) be neighbours. Now consider the ancestors of \( \mu \) and \( \nu \) at level \( l \). Their addresses will be \( \langle M_l, \ldots, M_{h-1} \rangle \) and \( \langle N_l, \ldots, N_{h-1} \rangle \) respectively. Then, since \( M_x \neq N_x \) for \( 0 \leq x \leq h-1 \), \( \mu \) and \( \nu \) can have no common ancestor nearer than the root. Therefore the shortest path between \( \mu \) and \( \nu \) passes through the link between their respective ancestors at level \( h-1 \) and has a length of \( 2h-1 \). ■
Lemma 3.2. In $HiC_{k,h}$ there exist at least two nodes, $\mu$ with address $M = (M_0, \ldots, M_{h-1})$ and $\nu$ with address $N = (N_0, \ldots, N_{h-1})$, such that $M_x \neq N_x$ for $0 \leq x \leq h-1$.

Proof. Nodes at level 0 have an address made up of $h$ digits in the range 1 to $k$. The number of unique addresses which can be generated with a sequence of $h$ digits, where each digit is in the range 1 to $k$, is $k^h$. The number of nodes at level 0 is $k^h$, and each node has a unique address. ■

Theorem 3.1. A hierarchical clique $HiC_{k,h}$ with $k > 1$ and $h > 0$ has diameter $D = 2h - 1$.

Proof. An $HiC_{k,h}$ is a $k$-ary tree of height $h$, modified with extra links. A complete tree of height $h$ has diameter $D = 2h$. Since the nodes at level $h-1$ form a clique the distance from any node at level $h-1$ to any other node at level $h-1$ is $d = 1$. Therefore the maximum possible distance between two nodes in an $HiC_{k,h}$ is $2h-1$. From Lemmas 3.1 and 3.2 there exists at least one pair of nodes $\mu, \nu$ with distance $d(\mu, \nu) = 2h - 1$. ■

Another important parameter of a multiprocessor interconnection network is the average distance between processors. The average inter-PE distance is often used in dynamic network topologies rather than the average inter-node distance, as it is a more useful indicator of performance. However, the symbol $\langle d \rangle$ is used to represent the average inter-PE distance, as it is frequently compared with the average internode distance of static networks.

Theorem 3.2. A hierarchical clique $HiC_{k,h}$ with $k > 1$ and $h > 0$ has average inter-PE distance

$$\langle d \rangle = \frac{2^{h-1}(1-k^h) + (2h+1)k^h - k^{h-1}}{k^h - 1}.$$
Proof. Observe that the HiC topology is symmetric from the point of view of
the leaves. Without loss of generality then, \( d \) can be determined by considering one
leaf node, \( \mu \). The distance from \( \mu \) to another leaf node \( \nu \) is \( d(\mu, \nu) \). We wish to
determine \( \sum d(\mu, \nu) \) for all \( \nu \) at level 0.

- The sum of distances to \( \mu \)'s \((k - 1)\) neighbours is \((k - 1)\).
- At level \( l \) \((0 < l < h)\), \( \mu \) has an ancestor \( u \), which is the least common ancestor
  of \( \mu \) and \((k - 1)k^{l-1}\) leaf nodes. The distance from any of these leaf nodes to
  \( \mu \) is \(2l\), so the sum of the distances equals \(2l(k - 1)k^{l-1}\). Considering all leaf
  nodes which have a least common ancestor, other than the root, with \( \mu \), the sum of
distances is \( \sum_{l=1}^{h-1} 2l(k - 1)k^{l-1} \).
- Each of \( u \)'s \((k - 1)\) neighbours at level \( l \) is the least common ancestor of one
  of \( \mu \)'s neighbours and \((k - 1)k^{l-1}\) leaf nodes. The distance from any of these
  leaf nodes to \( \mu \) is \(2l + 1\), so the sum of the distances equals \((k - 1)^{2}k^{l-1}(2l + 1)\).

Considering all leaf nodes which have a least common ancestor, other than the root,
with one of \( \mu \)'s neighbours, the sum of distances is \( \sum_{l=1}^{h-1} (k - 1)^{2}k^{l-1}(2l + 1) \).

These three cases cover all possible leaf nodes. Their partial results can be combined
to give the expression for the total sum of distances.

\[
\sum d(\mu, \nu) = (k - 1) + (k - 1) \sum_{l=1}^{h-1} 2lk^{l-1} + (k - 1)^2 \sum_{l=1}^{h-1} k^{l-1}(2l + 1) \hspace{1cm} (1)
\]

\[
= (k - 1)(k^{h-1}) + 2 \sum_{l=1}^{h-1} lk^{l} \hspace{1cm} (2)
\]

\[
= (k - 1)k^{h-1} + \frac{2((h - 1)k^{h+1} - hk^{h} + k)}{(k - 1)} \hspace{1cm} (3)
\]

\[
= -\frac{2}{(k - 1)}k^{h+1} + (2h + 1)k^{h} - k^{h-1} + \frac{2}{(k - 1)}k. \hspace{1cm} (4)
\]

The total number of distances is \( N_P - 1 = k^{h} - 1 \),

\[
\therefore d = -\frac{2}{(k - 1)}k^{h+1} + (2h + 1)k^{h} - k^{h-1} + \frac{2}{(k - 1)}k \frac{k^{h} - 1}{k^{h} - 1}
\]
Values of average inter-PE distance for various configurations of $HiC_{(k,h)}$ are plotted in Figure 2. The plots clearly show the reduced average distance which is obtained in a system with a given number of PEs by increasing the value of $k$. This improvement is gained at the cost of extra links and extra switch complexity, which naturally increases the cost of any system. To justify the extra cost, algorithms must perform better with the extra PE connectivity provided by the extra links.

![Graph showing average distance vs number of PEs for different values of k.](image)

**FIG. 2.** Average Distance between PEs in $HiC$

4. FAULT-TOLERANCE

In this section we study the fault-tolerance of the $HiC$ interconnection network. The connectivity, fault diameter, two-terminal reliability and average two-terminal
reliability of the HiC topology are determined. The results are compared with those of the binary hypercube. The binary hypercube is a fault-tolerant topology and its characteristics have been widely studied and reported [14, 6, 15]. This makes it an ideal topology to use as a benchmark for fault-tolerance. In general, any system with fault-tolerance greater than or near to that of the binary hypercube can be considered to have good fault-tolerance.

4.1. Connectivity

We first prove some theorems regarding basic graph theoretic properties of HiCs.

Theorem 4.1. The degree \( \delta(HiC_{(k,h)}) = k \).

Proof. \( \delta(G) = \min(\delta_{\mu}) : \forall \mu \in V(G). \) In \( HiC_{(k,h)} \), degree \( \delta_{\mu} = k \) when \( \mu \) is the root node or any of the leaf nodes and \( \delta_{\mu} = 2k \) when \( \mu \) is any other node. \( \blacksquare \)

Theorem 4.2. The connectivity \( \kappa(HiC_{(k,h)}) = k \).

Proof. In \( HiC_{(k,h)} \), \( k \) node-disjoint paths exist from any leaf \( \mu \) to any other node via \( \mu \)'s parent and \( k - 1 \) neighbours. The root node has \( k \) node-disjoint paths to any other node via its \( k \) children. Any node \( \nu \) at level \( l \), where \( 0 < l < h \), has \( k \) node-disjoint paths to descendants, or descendants of \( \nu \)'s neighbours, via \( \nu \)'s \( k - 1 \) neighbours and the appropriate one of \( \nu \)'s children. Node \( \nu \) has \( k \) node-disjoint paths to all other nodes via \( \nu \)'s parent and \( k - 1 \) neighbours. In \( HiC_{(k,h)} \), therefore, \( k \) node-disjoint paths exist between any two nodes. \( \blacksquare \)

The \( HiC_{(k,h)} \) is therefore \( (k - 1) \) node fault-tolerant.

Theorem 4.3. The link-connectivity \( \lambda(HiC_{(k,h)}) = k \).

Proof. In \( HiC_{(k,h)} \), \( k \) link-disjoint paths exist from any leaf \( \mu \) to any other node via \( \mu \)'s parent and \( k - 1 \) neighbours. The root node has \( k \) link-disjoint
paths to any other node via its \( k \) children. Any node \( \nu \) at level \( l \), where \( 0 < l < h \), has \( k \) link-disjoint paths to descendants, or descendants of \( \nu \)'s neighbours, via \( \nu \)'s \( k - 1 \) neighbours and the appropriate one of \( \nu \)'s children. Node \( \nu \) has \( k \) link-disjoint paths to all other nodes via \( \nu \)'s parent and \( k - 1 \) neighbours. In \( HiC_{(h,k)} \), therefore, \( k \) link-disjoint paths exist between any two nodes. □

The \( HiC_{(h,k)} \) is therefore \((k - 1)\) link fault-tolerant.

A more accurate reflection of the practical fault-tolerance of the network is given by determining the probability of any fault set of \( \kappa \) nodes or \( \lambda \) links disconnecting the network [5].

Theorem 4.4. The probability \( p \) of an \( HiC_{(h,k)} \) being disconnected by a \( k \)-node fault set is

\[
p = \frac{k^h(2^k - 2) + k^{(h-1)} - k(2^k - 2) - 1}{k - 1} \times \frac{(N - k)!k!}{N!}.
\]

Proof. The total number \( F \) of \( k \)-node fault sets possible is \( F = \binom{N}{k} \) which can be represented as \( F = \frac{N!}{(N-k)!k!} \). Let \( T_l \) be the number of \( k \)-node fault sets which can occur at level \( l \) which cause disconnection of the network. Let \( T = \sum_{l=0}^{h} T_l \) be the total number of \( k \)-node fault sets which cause disconnection of the network. At level \( l = h \) there is only the root node, therefore \( T_h = 0 \). At level \( l = h - 1 \) disconnection will occur only if all \( k \) nodes in a clique are faulty. There is only one clique at this level, therefore \( T_{(h-1)} = 1 \). At level \( l \), where \( 0 < l < (h - 1) \), disconnection of the network will occur if \( n \) \((0 < n \leq k)\) nodes in a clique are faulty, and the parents of the \((k - n)\) non-faulty nodes in the clique are faulty. The number of ways such a failure can occur at a particular clique is \( 2^k - 1 \). The total number of cliques for \( 0 < l < (h - 1) \) is \( \sum_{l=1}^{h-2} k^{(h-l-1)} = \frac{[k^{(h-1)}-k]}{[k-1]} \) for \( k \neq 1 \). Therefore \( \sum_{l=1}^{h-2} T_l = \frac{(2^k-1)[k^{(h-1)}-k]}{[k-1]} \). At level \( l = 0 \), disconnection of the network will occur if \( n \) \((0 < n < k)\) nodes in a clique are faulty, and the parents of the
\((k-n)\) non-faulty nodes in the clique are faulty. The number of ways such a failure can occur at a particular clique is \(2^k - 2\). The total number of cliques for \(l = 0\) is \(k^{(h-1)}\). Therefore \(T_0 = k^{(h-1)}(2^k - 2)\). Summing these terms we obtain:

\[
T = T_h + T_{(h-1)} + \sum_{i=1}^{h-2} T_i + T_0
\]

\[
= 1 + \frac{(2^k - 1)(k^{(h-1)} - k)}{(k - 1)} + k^{(h-1)}(2^k - 2)
\]

\[
= \frac{(k^{h-1} - 1)(2^h k - 2k + 1)}{k - 1}.
\]

The probability of any given \(k\) node fault set causing disconnection is given by 

\[
T/F.
\]

4.2. Fault-Diameter

From Theorem 4.2 we have \(\kappa(\text{HiC}_{(k,h)}) = k\). The fault diameter \(f(\text{HiC}_{(k,h)})\) is the largest diameter of the network in the presence of a fault set of \(k-1\) nodes. Determining \(f(\text{HiC}_{(k,h)})\) involves computing \(\max d(\mu, \nu)\) for each pair of PEs in \(\text{HiC}_{(k,h)}\) given any \(k - 1\) node fault set.

Let \(S(G)\) be the set of all pairs of nodes in \(V(G)\). As we are concerned with the ability of a network to provide communication between PEs, let \(S(\text{HiC}_{(k,h)})\) be the set of all pairs of leaf nodes in \(V(\text{HiC}_{(k,h)})\). The pairs of leaf nodes within \(S(\text{HiC}_{(k,h)})\) can be classified according to the distance \(d(\mu, \nu)\). We divide \(S(\text{HiC}_{(k,h)})\) into three classes and prove a lemma on each.

**Lemma 4.1.** If \(d(\mu, \nu) = 1\) there are \(k\) node and link disjoint paths between \(\mu\) and \(\nu\) of length at most \(d(\mu, \nu) + 2\).

**Proof.** Node \(\mu\) has address \(M = \langle M_0, \ldots, M_{h-1} \rangle\) and node \(\nu\) has address \(N = \langle N_0, \ldots, N_{h-1} \rangle\). Let \(u\) be a leaf node which is a neighbour of \(\mu\) and of \(\nu\).

1. There is one path of length 1:

\[
\mu \rightarrow \nu.
\]
2. There are $k - 2$ paths of length $2$:

$\mu \rightarrow$ one of $k - 2$ possible nodes $u \rightarrow \nu$.

3. There is one path of length $3$:

$\mu \rightarrow \mu's\ parent \rightarrow \nu's\ parent \rightarrow \nu$.

\[\]

**Lemma 4.2.** If $d(\mu, \nu) > 1$ and $d(\mu, \nu)$ is **O**DD there are $k$ node and link disjoint paths between $\mu$ and $\nu$ of length at most $d(\mu, \nu) + 1$.

**Proof.** Node $\mu$ has address $M = (M_0, \ldots, M_{k-1})$ and node $\nu$ has address $N = (N_0, \ldots, N_{k-1})$. Let node $u$ with address $U = (M_0, \ldots, M_{k-2}, N_{k-1})$ and node $v$ with address $V = (N_0, \ldots, N_{k-2}, M_{k-1})$ be two other leaf nodes. Then $u$ is a neighbour of $\mu$ and the least common ancestor of $u$ and $\nu$, denoted by $lca(u, \nu)$, exists at level $l$, where $0 < l < h$. Also, $v$ is a neighbour of $\nu$ and $lca(v, \mu)$ exists at level $l$, where $0 < l < h$. Let $u$ be any leaf node other than $u$ which is a neighbour of $\mu$. Let $v$ be a leaf node other than $v$ which is a neighbour of $\nu$ such that $lca(v, u)$ exists at level $l$.

1. There is one path of length $d(\mu, \nu)$:

$\mu \rightarrow lca(\mu, v) \rightarrow v \rightarrow \nu$.

2. There is one path of length $d(\mu, \nu)$:

$\mu \rightarrow u \rightarrow lca(u, \nu) \rightarrow \nu$.

3. There are $k - 2$ paths of length $d(\mu, \nu) + 1$, each including:

$\mu \rightarrow$ one of $k - 2$ possible nodes $u \rightarrow v \rightarrow \nu$.

\[\]
Proof. Distance $d(\mu, \nu)$ is EVEN if and only if $lca(\mu, \nu)$ exists at level $l$, where $1 < l < h$. Let $u$ be any leaf node which is a neighbour of $\mu$. Let $v$ be a leaf node which is a neighbour of $\nu$ such that $lca(\nu, u)$ exists at level $l$.

1. There is one path of length $d(\mu, \nu)$:
   \[ \mu \rightarrow lca(\mu, \nu) \rightarrow \nu. \]

2. There are $k - 1$ paths of length $d(\mu, \nu) + 2$ each including
   \[ \mu \rightarrow \text{one of } k - 1 \text{ possible nodes } u \rightarrow v \rightarrow \nu. \]

\[ \blacksquare \]

Lemma 4.4. In an $HiC_{(k, h)}$ with $h \geq 2$, any two leaf nodes have at least $k$ node disjoint paths between them of length $2h$ or less.

Proof. From Lemma 4.1, if $d(\mu, \nu) = 1$ there are $k$ node disjoint paths of maximum length $3$ between $\mu$ and $\nu$. For $h \geq 2$, $3 \leq 2h - 1$.

From Theorem 3.1, $D(HiC_{(k, h)}) = 2h - 1$, which implies that $D(HiC)$ is always ODD. Therefore $\max(d(\mu, \nu)) = D(HiC)$ if $d(\mu, \nu)$ is ODD. From Lemma 4.2, if $d(\mu, \nu)$ is ODD, there are $k$ node disjoint paths of maximum length $d(\mu, \nu) + 1$ between $\mu$ and $\nu$. Combining these two results, we see that nodes $\mu$ and $\nu$ with $d(\mu, \nu)$ ODD, have $k$ node disjoint paths of maximum length $2h$ between them.

From Lemma 4.3, if $d(\mu, \nu)$ is EVEN there are $k$ node disjoint paths between $\mu$ and $\nu$ of length at most $d(\mu, \nu) + 2$. But distance $d(\mu, \nu)$ is EVEN if and only if $lca(\mu, \nu)$ exists at level $l$, where $1 < l < h$. For $l < h$ path length $2l + 2$ is never greater than $2h$. Therefore there are $k$ node disjoint paths of maximum length $2h$.

These three cases include all possible pairs of leaf nodes, hence the proof. \[ \blacksquare \]

Theorem 4.5. A hierarchical clique $HiC_{(k, h)}$ has a fault diameter $f = 2h$ for $k > 2$ and $h > 1$. 

Draft January 12, 2000, 10:33am Draft
Proof. Let \( \mu \) and \( \nu \) be two leaf nodes at distance \( d(\mu, \nu) \). From Theorem 3.1 we know that at least one pair of nodes \( \mu, \nu \) exists with distance \( d(\mu, \nu) = D(HiC_{(k,h)}) = 2h - 1 \). If \( k > 2 \) then a fault set of \( k - 1 \) nodes can break all the paths between \( \mu \) and \( \nu \) of length \( 2h - 1 \). From Lemma 4.4 we see that a fault set of \( k - 1 \) nodes cannot increase \( d(\mu, \nu) \) to more than \( 2h \). The largest diameter in the presence of a \( k - 1 \) node fault set is therefore \( 2h \). 

In [7] two classes of graphs based on fault diameter were distinguished.

**Definition 4.1.** A class of graphs \( G_i \) is **strongly resilient** if, for all \( i \), there exists a constant \( t \) such that \( f(G) \leq D(G) + t \).

**Definition 4.2.** A class of graphs \( G_i \) is **weakly resilient** if, for all \( i \), there exists a constant \( t \) such that \( f(G) \leq D(G) \times t \).

The \( HiC_{(k,h)} \) is strongly resilient, since \( f = D + 1 \). This indicates that even under maximally faulty conditions the performance of the \( HiC_{(k,h)} \) will not be severely degraded.

In Figure 3 the fault diameter of the binary \( n \)-cube is compared with that of \( HiC_{(4,h)} \) and \( HiC_{(3,h)} \) for a range of network sizes. The fault diameter of the \( HiC_{(3,h)} \) increases most rapidly with increasing network size, followed by the \( n \)-cube. The \( HiC_{(4,h)} \) clearly has the lowest fault diameter for all network sizes considered. Given \( HiC_{(k1,h1)} \) and \( HiC_{(k2,h2)} \) with identical numbers of PEs, if \( k1 > k2 \) then \( f(HiC_{(k1,h1)}) < f(HiC_{(k2,h2)}) \).

### 4.3. Two-Terminal Reliability

The **two-terminal reliability** or **path reliability** \( R_2(\mu, \nu) \) between a pair of nodes \( \mu \) and \( \nu \) in \( V(G) \) is defined as the probability of finding a path entirely composed of operational links and nodes between \( \mu \) and \( \nu \) [3]. Let \( S(G) \) be the set of all pairs
of nodes in $V(G)$, then the two-terminal reliability $R_2(G)$ of graph $G$ is defined as 
$$(\min R_2(\mu, \nu)) : \forall (\mu, \nu) \in S(G).$$
As we are concerned with the ability of a network to provide communication between processors, the two-terminal reliability for the HiC deals only with leaf nodes. Let $S(HiC_{(k,h)})$ be the set of all pairs of leaf nodes in $HiC_{(k,h)}$. In this section a lower bound for $R_2(HiC_{(k,h)})$ will be determined.

From Theorem 4.3, there are at least $k$ link-disjoint paths between any given pair of leaf nodes $\mu$ and $\nu$ in $HiC_{(k,h)}$. Assume that link failures are statistically independent and occur randomly in time. A lower bound $[R_2(\mu, \nu)]$ can be determined by establishing the probability of at least one of the $k$ link-disjoint paths

\[ \text{Fault Diameter} \]
\[ \text{Number of PEs} \]

FIG. 3. The Fault-Diameter of n-cube and HiC.
between leaf nodes \( \mu \) and \( \nu \) being completely operational. Using Lemmas 4.1, 4.2 and 4.3, \( \min\{R_2(\mu, \nu)\} : \forall (\mu, \nu) \in S(HiC_{(k, h)}) \) can be obtained; a lower bound on \( R_2(HiC_{(k, h)}) \). Assume that all links have a probability \( p \) of being operational. Then the probability of any path of length \( l \) being operational is \( p^l \) and the probability of failure of such a path is \( 1 - p^l \). The probability of failure of \( i \) link disjoint paths, all of length \( l \), is given by \( (1 - p^l)^i \). Therefore the probability that at least one of \( i \) link disjoint paths of length \( l \) is operational is \( 1 - (1 - p^l)^i \).

**Theorem 4.6.** If \( h > 1 \) then \( [R_2(HiC_{(k, h)})] = 1 - (1 - p^{D(HiC)+1})^k \).

**Proof.** From Theorem 3.1, \( D(HiC_{(k, h)}) = 2h - 1 \), which implies that \( D(HiC) \) is always ODD. Therefore \( \max(d) = D(HiC) \) if \( d \) is ODD and \( \max(d) = D(HiC) - 1 \) if \( d \) is EVEN. From Lemma 4.3 we know that if \( d(\mu, \nu) \) is EVEN there are \( k \) link-disjoint paths between \( \mu \) and \( \nu \) of length at most \( d(\mu, \nu) + 2 \). Therefore a lower bound \( [R_2(\mu, \nu)] \) is given by \( (1 - (1 - p^{d(\mu, \nu)+2})^k) \). But \( \max(d) = D(HiC) - 1 \) if \( d \) is EVEN. Therefore \( \min(1 - (1 - p^{d+2})^k) = 1 - (1 - p^{D(HiC)+1})^k \).

From Lemma 4.2 we know that if \( d(\mu, \nu) > 1 \) is ODD there are \( k \) link-disjoint paths between \( \mu \) and \( \nu \) of length at most \( d(\mu, \nu) + 1 \). Therefore a lower bound \( [R_2(\mu, \nu)] \) is given by \( (1 - (1 - p^{d(\mu, \nu)+1})^k) \). But \( \max(d) = D(HiC) \) if \( d \) is ODD. Therefore \( \min(1 - (1 - p^{d+1})^k) = 1 - (1 - p^{D(HiC)+1})^k \).

A lower bound \( [R_2(HiC_{(k, h)})] \) is given by \( \min([R_2(\mu, \nu)]): \forall (\mu, \nu) \in S(HiC_{(k, h)}) \). Since \( \min([R_2(\mu, \nu)]) \) is given by \( (1 - (1 - p^{D(HiC)+1})^k) \) if \( d(\mu, \nu) \) is EVEN or if \( d(\mu, \nu) \) is ODD and greater than one, and if \( h > 1 \), we conclude that \( [R_2(HiC_{(k, h)})] \) is given by \( 1 - (1 - p^{D(HiC)+1})^k \). 

### 4.4. Average Two-Terminal Reliability

While two-terminal reliability can give an indication of the likelihood of failure of communications between specific pairs of nodes, or even classes of pairs of nodes,
it gives no indication of the average path reliability of a pair of nodes in the graph.

A graph with a few short, reliable links and many long, unreliable ones is indistinguishable from a graph with many short, reliable links and few long, unreliable ones. Average two-terminal reliability overcomes this shortcoming. Average two-terminal reliability \( \overline{R}_2(G) \) is defined as \( \frac{1}{|V(G)|} \sum R_2(\mu, \nu) : \forall (\mu, \nu) \in S(G) \) where \( |S(G)| = \left( \frac{|V(G)|}{2} \right) \). In this section we derive a lower bound on average two-terminal reliability for the HiC\([k, k]\). A lower bound on the average two-terminal reliability of the binary hypercube will also be derived, and compared with that of the HiC for a range of system sizes.

A network is symmetric if it is isomorphic to itself with any node labelled as the origin. A symmetric network appears the same if viewed from the perspective of any node. Therefore, all values of \( R_2(\mu, \nu) \) can be determined by considering only those sets of nodes in \( S(G) \) which contain a given node \( \mu \). Define \( S(G, \mu) \subset S(G) \) as the set of all pairs of nodes in \( V(G) \) which contain node \( \mu \). For symmetric networks, then, we can redefine \( \overline{R}_2(G) \) as \( \frac{1}{|V(G)|} \sum R_2(\mu, \nu) : \forall (\mu, \nu) \in S(G, \mu) \). Symmetric networks have a further property however: There is a constant \( k \) such that \( R_2(\mu, \nu) = k : \forall (\mu, \nu) \in S(G) \) with a given distance \( d \).

Define \( S(G, \mu, d) \subset S(G) \) as the set of all pairs of nodes in \( V(G) \) which contain node \( \mu \) and have a given value of distance \( d \). We can now redefine \( \overline{R}_2(G) \) as

\[
\sum \left( \frac{(R_2(\mu, \nu) : \forall (\mu, \nu) \in S(G, \mu, d)) \times |S(G, \mu, d)|}{|V(G)| - 1} : \forall S(G, \mu, d) \in S(G, \mu) \right), \tag{8}
\]

Since for the HiC we are concerned only with the PEs, we can replace \( |V(G)| - 1 \) with \( k^h - 1 \).

Recall that Theorem 4.6 established lower bounds \( [R_2(\mu, \nu)] \) for all pairs of distinct nodes \( (\mu, \nu) \in S(HiC, \mu) \).

1. \( [R_2(\mu, \nu)] \) is \( 1 - (1 - p^3)^h : \forall (\mu, \nu) \in S(HiC, \mu, d : 1) \).
2. \( [R_2(\mu, \nu)] \) is \( 1 - (1 - p^{d+2})^h : \forall (\mu, \nu) \in S(HiC, \mu, d : EVEN) \).
3. \( |R_2(\mu, \nu)| = (1 - (1 - p^{d+1})^k) : \forall (\mu, \nu) \in S(HiC, \mu, d : ODD > 1). \)

It remains to determine \( |S(HiC, \mu, d)| \) for each \( S(HiC, \mu, d) \in S(HiC, \mu) \).

1. If \( d(\mu, \nu) = 1 \) then \( \mu \) and \( \nu \) are neighbours, and

\[
|S(HiC, \mu, d : 1)| = k - 1.
\]

2. If \( d(\mu, \nu) \) is even then \( \mu \) and \( \nu \) share a common ancestor, and

\[
|S(HiC, \mu, d : EVEN)| = k\left(\frac{d}{2}\right) - k\left(\frac{d}{2} - 1\right).
\]

3. If \( d(\mu, \nu) > 1 \) is odd then \( \mu \) and \( \nu \) are not in the same clique and do not share a common ancestor, so

\[
|S(HiC, \mu, d : ODD > 1)| = (k\frac{d+1}{2} - k\frac{d+3}{2})(k - 1).
\]

Thus a lower bound \( \left| \overline{R_2(HiC_{(k,h)})} \right| \) is made up of the sum of three terms. The first term comes from node pairs with a distance of 1.

\[
Term1 = \frac{(k - 1)}{(k^h - 1)}(1 - (1 - p^3)^k)
\]

The second term comes from node pairs with an even distance.

\[
Term2 = \sum_{\frac{d}{2} - 1}^{h-1} (1 - (1 - p^{d+2})^k) \frac{k\left(\frac{d}{2}\right) - k\left(\frac{d}{2} - 1\right)}{k^h - 1}
\]

\[
= \frac{(k - 1)}{k(k^h - 1)} \sum_{\frac{d}{2} - 1}^{h-1} k\left(\frac{d}{2}\right)(1 - (1 - p^{d+2})^k)
\]

The third term comes from node pairs with an odd distance greater than 1.

\[
Term3 = \sum_{\frac{d+1}{2} - 1}^{\frac{d+1}{2} - 1} (1 - (1 - p^{d+1})^k) \frac{k\left(\frac{d+1}{2}\right) - k\left(\frac{d+3}{2}\right)(k - 1)}{k^h - 1}
\]

\[
= \frac{(k - 1)^2}{k(k^h - 1)} \sum_{\frac{d+1}{2} - 1}^{h-1} k\left(\frac{d+1}{2}\right)(1 - (1 - p^{d+1})^k)
\]
\[ [R_2(HiC_{(h,\lambda)})] \text{ is } \text{Term}1 + \text{Term}2 + \text{Term}3. \]

We now determine a lower bound \([R_2(Q)]\) on the average two-terminal reliability for a hypercube \(Q\) of dimension \(n\). Since the hypercube is symmetric, the expression for \(R_2(G)\) given in (8) is applicable. The total number of node pairs considered is given by \(|V(Q)| - 1 = 2^n - 1\). We can also determine that \(|S(Q, \mu, d)| = \frac{n!}{(n-d)! \times d!} \).

A lower bound on the two terminal reliability \([R_2(Q)]\) was determined in [5]. If the probability of a link being operational is \(p\), then

\[
[R_2(Q)] = \min(1 - ((1 - p^d)^{d}(1 - p^{(d+2)\text{(n-d)}})))
\]

**FIG. 4.** The average two-terminal reliability of hypercube and HiC.
over all distances $0 < d \leq n$. Therefore a lower bound is given by

$$[R_d(Q)] = \frac{n!}{(2^n - 1)} \sum_{d=1}^{n} \frac{1 - ((1 - p^d)(1 - p^{d+2})(n-d))}{(n - d)! \times d!}$$

$$= \frac{2^n}{2^n - 1} - \frac{1}{2^n - 1} \sum_{d=1}^{n} \binom{n}{d} (1 - p^d)(1 - p^{d+2})^{n-d},$$

(13)

(14)

4.5. Comparison

The lower bounds determined for the average two-terminal reliability of hypercube, $HiC_{(4,k)}$ and $HiC_{(3,k)}$ are plotted against the number of PE’s in Figure 4. The lower bound on the two-terminal reliability of the hypercube rises with network dimension. This is as expected, since the connectivity and node degree of a hypercube also increase with dimension. The $HiC_{(k,k)}$ network, by contrast, has fixed connectivity and node degree, consequently the lower bound decreases with network size. The rate of decrease is much lower in the $HiC_{(4,k)}$ than in the $HiC_{(3,k)}$.

Further increasing the value of $k$ leads to higher bounds for the value of average two-terminal reliability, but at the cost of increased node degree. An $HiC_{(4,4)}$ has 256 PEs, node degree of 8 and an average two-terminal reliability of over 0.9 with an operational link probability of $p = 0.9$. With the same operational link probability, a hypercube of the same size has an average two-terminal reliability of nearly 1 but node degree of 16.

5. MESSAGE ROUTING ALGORITHMS

5.1. One to One Communication

PEs and SEs handle message routing in different ways. PEs either source or receive messages, SEs serve only as intermediate nodes for PE to PE communication.

Let the address of the current node be $\langle C_l, \ldots, C_{h-1} \rangle$ where $(0 \leq l < h)$ and the destination address be $\langle D_0, \ldots, D_{h-1} \rangle$.

A PE sending a message compares $\langle C_0, \ldots, C_{h-2} \rangle$ with $\langle D_0, \ldots, D_{h-2} \rangle$. If they are not the same the message is routed to the PE’s parent. If they are the same...
then the destination PE is a neighbour and the message is passed directly. This is expressed formally in Algorithm 1.

```plaintext
if \( \langle C_0, \ldots, C_{h-2} \rangle = \langle D_0, \ldots, D_{h-2} \rangle \) then
   Route message to neighbouring PE \( \langle D_0, \ldots, D_{h-1} \rangle \)
else
   Route message to parent

Algorithm 1: HiC PE message routing algorithm
```

A SE at level \( l \) has address \( \langle C_l, \ldots, C_{h-1} \rangle \) for \( 0 < l < h - 1 \). Upon receipt of a message a SE compares its own address with the corresponding digits from the destination node address. If they are the same the SE is an ancestor of the destination node, and the message can be routed to the child node which is either the destination or an ancestor of the destination. If the addresses differ only in the rightmost digit, then a neighbour of the SE is an ancestor of the destination node, and the message is routed via that neighbour. In any other case the message is routed further up the hierarchy, to the SEs’ parent. An SE at level \( l = (h - 1) \) has address \( \langle C_l \rangle \). If \( \langle C_l \rangle = \langle D_l \rangle \) then the SE is an ancestor of the destination node, and the message is routed to the child node which is the destination or an ancestor of the destination. Otherwise the message is routed to the SE’s neighbour which is an ancestor of the destination node. This is expressed formally in Algorithm 2.

### 5.2. One to Many Communication

The HiC broadcast algorithms are very simple. The broadcast algorithm used by the source PE, with address \( \langle S_0, \ldots, S_{h-1} \rangle \), is trivial. The source PE routes the message only to its parent. For a SE at level \( l \), where \( 0 < l \leq h - 1 \), the broadcast algorithm is shown in Algorithm 3. Note that the message is transmitted to neighbours only at level \( l = h - 1 \). At all other levels only parent or child links are used. This is to prevent PEs from receiving duplicate messages.
if \(0 < l < h - 1\) then

if \(\langle C_l, \ldots, C_{h-1}\rangle = \langle D_l, \ldots, D_{h-1}\rangle\) then

Route message to child \(\langle D_{l-1}, \ldots, D_{h-1}\rangle\)

else

if \(\langle C_l, \ldots, C_{h-2}\rangle = \langle D_l, \ldots, D_{h-2}\rangle\) then

Route message to neighbouring SE \(\langle D_l, \ldots, D_{h-1}\rangle\)

else

Route message to parent

else

if \(l = h - 1\) then

if \(\langle C_l\rangle = \langle D_l\rangle\) then

Route message to child \(\langle D_{l-1}, D_l\rangle\)

else

Route message to neighbouring SE \(\langle D_l\rangle\)

Algorithm 2: HiC SE message routing algorithm

if \(l = h - 1\) then

Send message to all neighbours

else

if \(\langle C_l, \ldots, C_{h-1}\rangle = \langle S_l, \ldots, S_{h-1}\rangle\) then

Send message to parent and all children such that

\(\langle C_{l-1}, \ldots, C_{h-1}\rangle \neq \langle S_{l-1}, \ldots, S_{h-1}\rangle\)

else

Send message to all children

Algorithm 3: HiC SE broadcast algorithm

6. NETWORK EMBEDDINGS

The ability of a parallel computer’s interconnection network to emulate other networks is important if it is to effectively employ algorithms and data structures
developed for different parallel architectures. The quality of such an emulation can be determined by studying embeddings of one topology into another. Let \( G \) and \( H \) be undirected graphs. \( G \) represents a guest graph and \( H \) represents a host graph. Using the terminology of [11] an embedding of \( G \) into \( H \) is a mapping \( \mathcal{F} \) from the nodes of \( G \) to the nodes of \( H \). The mapping from any node \( g \) in \( G \) to a node \( h \) in \( H \) is represented by \( \mathcal{F}(g) = (h) \). The dilation of an embedding \( \mathcal{F} \) is the maximum distance in the host between the images of adjacent guest nodes. The expansion of the embedding \( \mathcal{F} \) is the ratio of the number of nodes in the host graph to the number of nodes in the guest graph, i.e. \( |V(H)|/|V(G)| \). The number of edges of \( G \) routed through edge \( e \in E(H) \) in embedding \( \mathcal{F} \) is \( c(e) \). The edge congestion of \( \mathcal{F} \) is \( \max(c(e)) : \forall e \in E(H) \).

Binary structures such as binary trees, binary hypercubes or meshes, map most naturally onto the \( HiC(k,h) \) when \( k \) is a power of two. In the following work, all mappings are made to \( HiC(4,h) \).

6.1. Binary Trees

For binary trees, two different embedding strategies are possible. If the binary tree is considered as a dynamic network with only leaf nodes as PEs, then SEs of the binary tree can be mapped to SEs of the \( HiC \). Alternatively if considered as a task graph or data structure, then every node must be mapped to a PE in the host \( HiC \). In the following work both cases are considered.

First consider the case of a dynamic tree network. For any \( t \geq 0 \), the complete binary tree \( B(t) \) of height \( t \) can be embedded in \( HiC(4,h) \) where \( h = \lfloor \frac{t+2}{2} \rfloor \). For binary tree \( B(t) \), the root node is at level zero, the leaf nodes at level \( t \). The root node is represented by \((0,0)\). The children of the root node are represented by \((1,0)\) and \((1,1)\). In a binary tree, \( b_j \) is a node at level \( j \), where \( t \geq j \geq 1 \). For \( HiC(4,h) \) the address of a node \( \mu \) at level \( l \) is represented by \( M = \langle M_l, \ldots, M_{h-1} \rangle \).
the address of a node $\nu$ at level $l$ is represented by $N = \langle N_l, \ldots, N_{h-1} \rangle$. The embedding is described in Algorithm 4.

$B(t)$ is a subgraph of $HiC(4,h)$, and can therefore be embedded with both dilation and edge congestion of one. The expansion is $\frac{1}{2^{h+1}} \sum_{l=0}^{h} 4^{h-l} = \frac{4^{h+1}-1}{3(2^{h+1})}.

\begin{align*}
\mathcal{F}(0, 0) &\equiv (1) \\
\mathcal{F}(1, 0) &\equiv (2) \\
\mathcal{F}(1, 1) &\equiv (3) \\
\text{for } j = 1 \rightarrow t \text{ do} \\
\quad \text{if } j \text{ is ODD then} \\
\quad \quad \text{if } \mathcal{F}(b_j) \equiv \mu \text{ such that } M_{h-1} \text{ is ODD then} \\
\quad \quad \quad \mathcal{F}(b_j'\text{'s first child}) \equiv \nu \text{ such that } N = \langle M_{h-1}, M \rangle \\
\quad \quad \quad \mathcal{F}(b_j'\text{'s second child}) \equiv \nu \text{ such that } N = \langle (M_{h-1} + 1), M \rangle \\
\quad \quad \else \\
\quad \quad \quad \mathcal{F}(b_j'\text{'s first child}) \equiv \nu \text{ such that } N = \langle M_{h-1}, M \rangle \\
\quad \quad \quad \mathcal{F}(b_j'\text{'s second child}) \equiv \nu \text{ such that } N = \langle (M_{h-1} - 1), M \rangle \\
\quad \quad \else \\
\quad \quad \quad \text{if } \mathcal{F}(b_j) \equiv \mu \text{ such that } M_{h-1} \text{ is EVEN then} \\
\quad \quad \quad \mathcal{F}(b_j'\text{'s first child}) \equiv \nu \text{ such that } N = \langle M_{h-1}, M \rangle \\
\quad \quad \quad \mathcal{F}(b_j'\text{'s second child}) \equiv \nu \text{ such that } N = \langle (M_{h-1} - 1), M \rangle \\
\quad \quad \else \\
\quad \quad \quad \mathcal{F}(b_j'\text{'s first child}) \equiv \nu \text{ such that } N = \langle M_{h-1}, M \rangle \\
\quad \quad \quad \mathcal{F}(b_j'\text{'s second child}) \equiv \nu \text{ such that } N = \langle (M_{h-1} - 1), M \rangle \\
\end{align*}

Algorithm 4: First Binary Tree Embedding

Now consider the case of a binary tree structure where each node of the tree must be mapped to a PE in a $HiC$. The embedding is described in Algorithm 5. A sample embedding is shown in Figure 5.
FIG. 5. An embedding of a binary tree.
\[ \mathcal{F}(0, 0) \equiv \mu \text{ with address } M = \langle M_0, \ldots, M_{h-1} \rangle \text{ such that } M_x = 1 \text{ for } (0 \leq x \leq h-1). \]
\[ \mathcal{F}(1, 0) \equiv \nu \text{ such that } N = \langle M_0, \ldots, M_{h-2}, 2 \rangle \]
\[ \mathcal{F}(1, 1) \equiv \nu \text{ such that } N = \langle M_0, \ldots, M_{h-2}, 4 \rangle \]
\[ \mathcal{F}(b_1\text{'s first child}) \equiv \nu \text{ such that } N = \langle (M_0 + 2), \ldots, M_{h-1} \rangle \]
\[ \mathcal{F}(b_1\text{'s second child}) \equiv \nu \text{ such that } N = \langle (M_0 + 2), \ldots, (M_{h-1} - 1) \rangle \]

\textbf{for} \( j = 2 \rightarrow t \) \textbf{do}

\hspace{1em} \textbf{if} \( j \) is EVEN \textbf{then}

\hspace{2em} \[ \mathcal{F}(b_j\text{'s first child}) \equiv \nu \text{ such that } N = \langle M_0, \ldots, (M_{(j/2)-1}, 1), \ldots, M_{h-1} \rangle \]

\hspace{2em} \[ \mathcal{F}(b_j\text{'s second child}) \equiv \nu \text{ such that } N = \langle M_0, \ldots, (M_{(j/2)-1} + 1), \ldots, M_{h-1} \rangle \]

\hspace{1em} \textbf{else}

\hspace{2em} \[ \mathcal{F}(b_j\text{'s first child}) \equiv \nu \text{ such that } N = \langle M_0, \ldots, (M_{(j/2)-1} - 1), (M_{(j/2)} + 2), \ldots, M_{h-1} \rangle \]

\hspace{2em} \[ \mathcal{F}(b_j\text{'s second child}) \equiv \nu \text{ such that } N = \langle M_0, \ldots, (M_{(j/2)} + 2), \ldots, M_{h-1} \rangle \]

\textbf{Algorithm 5: Second Binary Tree Embedding}

This embedding has:

\begin{align*}
\text{dilation} & \quad 2h - 1 \\
\text{expansion} & \quad \frac{2^{h+1}-1}{2^{h+1}+1} \\
\text{edge congestion} & \quad \begin{cases} 
4^t & : \quad t < 4 \\
4^t-h & : \quad t \ \text{EVEN} \geq 4 \\
3 \times 4^{t-h-1} & : \quad t \ \text{ODD} \geq 4.
\end{cases}
\end{align*}

\subsection*{6.2. Binary Hypercubes}

A binary hypercube \( Q(n) \) can be embedded into \( HiC_{4, \left\lfloor \frac{n+1}{2} \right\rfloor} \). For any \( Q(n) \) the address of a node \( q \) can be represented as a binary string \( (q_{n-1}, \ldots, q_0) \), [11]. For any binary string \( (q_x, \ldots, q_{x-y}) \), where \( y \leq x \), we represent the decimal value of the binary number \( (q_x, \ldots, q_{x-y}) \) by \( (q_x, \ldots, q_{x-y})_{10} \). For \( HiC_{4,h} \) the address of

\textbf{DRAFT January 12, 2000, 10:33am DRAFT}
a leaf node $\mu$ is represented by $M = \langle M_0, \ldots, M_{h-1} \rangle$. The embedding is described in Algorithm 6.

```
for all $q \in Q(n)$ do
    if $n$ is EVEN then
        $F(q) \equiv \mu$
        for $i = 0$ to $\lceil n/2 - 1 \rceil$ do
            $M_i = (q_{n-2i}, q_{n-2i-1})_{10} + 1$
    else
        $F(q) \equiv \mu$
        $M_{h-1} = (q_1, q_0)_{10} + 1$
        $M_{h-2} = (q_{n-1})_{10} + 1$
        for $i = 1$ to $\lceil n/2 - 1 \rceil$ do
            $M_{i-1} = (q_{n-2i}, q_{n-2i-1})_{10} + 1$

Algorithm 6: Binary Hypercube Embedding
```

This embedding has:

dilation $2(h - 1)$
edge congestion $2^{n-3}$
expansion $\frac{4^{n+1} - 1}{3.2^{n-1}} = \begin{cases} 1 & n \text{ EVEN} \\ 2 & n \text{ ODD}. \end{cases}$

The dilation of this embedding is derived in the following way. A node $q$ of $Q(n)$ has address $(q_{n-1}, \ldots, q_0)$. If a node $p$ of $Q(n)$ is one of the $n$ nodes directly connected to $q$ then $p$’s address differs from that of $q$ in that one bit $q_x$ is replaced by $q_x$, where $0 \leq x \leq n - 1$. From Algorithm 6 it is clear that, if $F(q) \equiv \mu$ and $F(p) \equiv \nu$, then the address of $\nu$ differs from that of $\mu$ in that one digit $M_y$ is replaced by $M_y + 1$, where $0 \leq y \leq h - 1$. If $y = h - 1$ then $\mu$ and $\nu$ are neighbours. If $0 \leq y < h - 1$ then $\text{lca}(\mu, \nu)$ occurs at level $y + 1$. The maximum distance between adjacent nodes of $Q(n)$ is therefore $2(h - 2 + 1) = 2(h - 1)$. 
6.3. Two Dimensional Meshes

A mesh $M$ of dimension $2^h \times 2^h$ can be embedded into $HiC(4,h)$. A node of mesh $M$ is identified as $(x,y)$, where $0 \leq x, y < 2^h$. For $HiC(4,h)$ the address of a leaf node $\mu$ at level $l$ is represented by $M = (M_0, \ldots, M_{h-1})$. The embedding is described in Algorithm 7. A sample embedding is shown in Figure 6.

This embedding has dilation $2(h - 1)$ and edge congestion $2^{h-1}$.

![Algorithm 7: 2D mesh Embedding](image)

**Algorithm 7:** 2D mesh Embedding

**FIG. 6.** An embedding of a 2D mesh.
6.4. Embeddings into the FatHiC

In all the embeddings considered above the edge congestion grows rapidly. This indicates congestion will occur in higher level communication channels, under uniform communication conditions, with these embeddings into the HiC. This problem has already been extensively studied, since this is possibly the most fundamental problem with tree based networks. A well accepted solution has been proposed, the Fat tree of [9]. In order to prevent congestion in the upper levels dominating communication time for regular algorithms involving a large proportion of PEs transmitting across the diameter of the network simultaneously, the bandwidth of links in upper layers is increased. Applying the same principle, define a FatHiC\(_{(k,h)}\) as being identical with an HiC\(_{(k,h)}\), except that the bandwidth of parent-child links increases with the level of the nodes incident to them. The rate of increase of bandwidth required is dictated by the cost and performance requirements of the system; the higher the bandwidth of the links the higher the expected performance, but at a higher cost. While considering embeddings into the FatHiC\(_{(k,h)}\) the link bandwidth will be assumed to increase by a factor of \(k\) at each level of the network. For example, a FatHiC\(_{(4,3)}\) would have links of bandwidth \(B\) from leaf nodes at level zero to their parents at level one, links of bandwidth \(4B\) from nodes at level one to their parents at level two, and links of bandwidth \(16B\) from nodes at level two to the root node. Assuming that the cost of the extra bandwidth is approximated by representing each link of bandwidth \(kB\) as \(k\) separate links, the number of links in this version of a FatHiC\(_{(k,h)}\) is \(L = k^h \left(\frac{k}{2} + h\right) - \frac{h}{2}\).

Since FatHiC\(_{(k,h)}\) is topologically identical to HiC\(_{(k,h)}\) any network which can be embedded into HiC\(_{(k,h)}\) can be embedded into FatHiC\(_{(k,h)}\) in exactly the same manner, with the same dilation and expansion. The congestion may be reduced. All embeddings considered in the previous sections were into HiC\(_{(4,3)}\); these same embeddings are used here into FatHiC\(_{(4,3)}\), and the congestion is determined.
Considering the second binary tree embedding of Section 6.1, the congestion of the embedding is 3 for \( t > 2 \).

Considering the binary hypercube embedding of Section 6.2, the congestion of the embedding is 2.

Considering the two dimensional mesh embedding of Section 6.3, the congestion of the embedding is 1.

## 6.5. Embedding Quality

In order to provide an indication of the relative quality of these embeddings, Table 1 shows embedding parameters for \( HiC(\{k,h\}) \), \( FatHiC(\{k,h\}) \) and another hierarchical network, the generalised fat trees \( GFT(\{h,m,w\}) \) [13]. All the results given here are for \( GFT(\{h,2,2\}) \). This configuration has a total number of nodes (PEs and SEs) \( N(GFT) = (h + 1) \times 2^h \) and a number of links \( L(GFT) = h(2^{h+1}) \).

For each of the guest graphs studied, the dilation and expansion of the embeddings into each of the host graphs are of the same order. In general, the dilation of the \( GFT \) is higher by some small factor, indicating possible higher latency for communications between mapped nodes. For each embedding the edge congestion of the \( HiC \) grows rapidly, the edge congestion of the \( GFT \) is either constant or grows slowly, while the edge congestion of the \( FatHiC \) is constant for each case. This indicates congestion would occur rapidly under uniform communication conditions with these embeddings into the \( HiC \). Such a result is not surprising, both the \( FatHiC \) used and the \( GFT(\{h,2,2\}) \) maintain constant bandwidth between all levels of the hierarchy while the bandwidth of the \( HiC \) decreases in the upper levels of the hierarchy. However, the potential performance benefit yielded by the lower edge congestion of the \( FatHiC \) and \( GFT \) must be weighed against the cost of the extra hardware required to achieve it. For example, a binary hypercube of order 6 maps onto \( HiC(\{4,3\}) \), \( FatHiC(\{4,3\}) \) or \( GFT(\{6,2,2\}) \). While each of these topologies has 64 PEs, the \( HiC(\{4,3\}) \) and \( FatHiC(\{4,3\}) \) have only 21 SEs, while the \( GFT(\{6,2,2\}) \) has

---

**DRAFT** January 12, 2000, 10:33am **DRAFT**
TABLE 1

<table>
<thead>
<tr>
<th>Guest graph</th>
<th>Host graph</th>
<th>Dilation</th>
<th>Expansion</th>
<th>Edge Congestion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binary Tree</td>
<td>HiC(_{(4, 1^{\frac{2h}{t}})})</td>
<td>(2(h - 1))</td>
<td>(\approx 1\ or 2)</td>
<td>(4^{(t-h)}\ or 3 \times 4^{(t-h-1)})</td>
</tr>
<tr>
<td></td>
<td>FatHiC(_{(4, 1^{\frac{2h}{t}})})</td>
<td>(2(h - 1))</td>
<td>(\approx 1\ or 2)</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>GFT(_{(t+2, 2, 2)})</td>
<td>(2(t + 1))</td>
<td>(\approx 1)</td>
<td>3</td>
</tr>
<tr>
<td>Hypercube</td>
<td>HiC(_{(4, \frac{n+1}{2}})</td>
<td>(2(h - 1))</td>
<td>1 or 2</td>
<td>(2^{n-3})</td>
</tr>
<tr>
<td></td>
<td>FatHiC(_{(4, \frac{n+1}{2}})</td>
<td>(2(h - 1))</td>
<td>1 or 2</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>GFT(_{(n, 2, 2)})</td>
<td>2(n)</td>
<td>1</td>
<td>([n/2])</td>
</tr>
<tr>
<td>2D-mesh</td>
<td>HiC(_{(4, h)})</td>
<td>(2(h - 1))</td>
<td>1</td>
<td>(2^{h-1})</td>
</tr>
<tr>
<td>2(^h) \times 2(^h)</td>
<td>FatHiC(_{(4, h)})</td>
<td>(2(h - 1))</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>GFT(_{(2h, 2, 2)})</td>
<td>4(h)</td>
<td>1</td>
<td>3</td>
</tr>
</tbody>
</table>

384 SEs. Similarly, the HiC\(_{(4, 3)}\) requires a total of 210 links, the FatHiC\(_{(4, 3)}\) has 318 links, while the GFT\(_{(6, 2, 2)}\) uses 768 links. Of course using different members of either family of networks would yield different results for these mappings. In general the HiC emphasises local communication over global communication and allows excellent, cost effective local connectivity at the cost of poorer global performance. The FatHiC has superior global communication and allows higher quality mappings than the GFT, with fewer switches and less links. The FatHiC is clearly the best of the hierarchical networks considered, offering high quality embeddings for reasonable cost.

7. PERFORMANCE OF THE HiC

The performance of an interconnection network under load is commonly assessed in terms of two parameters, throughput and latency.
Throughput is defined as the sustained data delivery rate given an applied load. While maximum potential throughput is obviously important, the saturation point and the behaviour after saturation are also significant. Saturation can occur at different load levels depending on traffic patterns and represents the maximum achievable throughput for a given traffic pattern. The behaviour of the network when the applied load exceeds the saturation point is important, as instability can result leading to degraded performance. It is desirable for the throughput to remain stable after saturation, whether due to bursty or sustained demand.

The network latency is the average time spent by a message in the network. It does not include source queueing delay.

7.1. Traffic Patterns

In our simulations each PE generates messages independently according to a uniform distribution and the destinations are chosen according to the following traffic patterns.

- **Uniform Traffic.** Destinations are chosen randomly, with equal probability between PEs.

- **Localised Traffic.** Destinations are chosen according to a weighted probability: 25% of traffic is addressed to a neighbour, 25% of traffic is directed to PEs at a distance of two or three, 25% of traffic is directed to PEs at a distance of four or five and 25% of traffic is directed to PEs at a distance of six or seven.

The uniform traffic pattern is a commonly used benchmark in network routing studies. The localised traffic pattern illustrates the capabilities of the network under the sort of traffic which occurs in many real problems.

7.2. Simulation Results

We simulate the behaviour of a $HiC_{(4,4)}$ with 256 PEs. The uni-directional bandwidth of the links equals the message length. For simplicity store and forward
routing was used, though more sophisticated routing techniques could as well be applied. Each simulation runs for 3500 cycles in order to reach steady state, then performance data is collected for 1500 cycles. The simulation results are presented according to the Chaos Normal Form\(^1\) (CNF). The HiC topology is not bisection bandwidth limited, so normalising to the bisection bandwidth limit is not useful. Rather we calculate the injection rate which causes saturation of the links between neighbours in the top level clique under uniform random traffic. Under uniform random traffic conditions, 75% of the messages from a PE are passed up to the top level clique to be rerouted to their destination. Of the messages arriving at a top level NC, 25% are routed to each neighbour, the remaining 25% are routed down to child nodes. For an HiC\(_{(4,4)}\) then, if each PE injects one message per cycle there are twelve messages per cycle passing through each link in the top level clique. For bandwidth of one message per cycle the load which causes saturation of the top level links is 0.083 messages injected per PE per cycle. We normalise all load and throughput figures to this limit.

7.2.1. Uniform Traffic

The normalised throughput equals the normalised load until the top level links saturate. This is an optimal result. The latency also remains very low until saturation occurs. After saturation although the rate of increase of throughput declines, the spare capacity of lower levels in the network ensure that throughput continues to increase. The latency rises sharply after saturation, but the rate of increase levels off. This can be attributed to the increasing proportion of messages being delivered locally, while the number of messages being delivered over greater distances remains static. A further drop in the rate of increase of throughput occurs when the normalised load exceeds three. This is perfectly in accordance with expectation, as

the links leading up to the second top level NCs also begin to saturate at this load level. Once again latency begins to increase, but the system remains stable. Note that our simulation did not include flow control. Thus for loads which cause saturation, average latency will continue to rise as long as the load is maintained and the buffers do not overflow. The figures given are indicative of the relative levels of latency at the corresponding loads, but absolute values depend on the number of cycles during which measurements are made.

![Graph](image1.png)

**FIG. 7.** Normalised throughput vs normalised applied load (uniform traffic)

![Graph](image2.png)

**FIG. 8.** Network latency vs normalised applied load (uniform traffic)
7.2.2. Localised Traffic

Once again the normalised throughput equals the normalised load until the top level links saturate, at a normalised load of three, which is optimal. Latency remains very low until just before saturation. After saturation, however, throughput continues to increase at only a slightly reduced rate. As in the previous case the spare capacity in the lower levels of the network allow for continued increase, but here the majority of traffic is routed locally, so the saturation of the top level has relatively less impact. Also, the average latency does not increase as rapidly as in the case of the uniform pattern, since a greater proportion of messages are not being delayed. As with the uniform case, the effects of lower level links saturating can be observed as load continues to increase. The system remains stable under all conditions.

![Graph](image_url)

**FIG. 9.** Normalised throughput vs normalised applied load (local traffic)

8. DISCUSSION AND CONCLUSION

We presented the HiC interconnection network for multiprocessor systems. This network, a combination of the tree and fully-connected networks, possesses topological and architectural properties superior to most other hierarchical networks in a variety of problem domains. Some features of the hierarchical cliques include:
1) flexibility in terms of degree of connectivity and the height of the tree; 2) fault tolerance, 3) the ability to make maximum use of communication locality in parallel algorithms, 4) logarithmic diameter, 5) possess versatile embedding properties. In addition, the proposed addressing scheme facilitates self routing and broadcasting.

We have demonstrated the performance of HiCs under various communication patterns. The HiC is particularly suited to problems with cluster-like communication patterns. We are investigating implementation aspects of subgraph matching algorithms on the HiC.

ACKNOWLEDGMENT

The authors would like to thank Dr J. Simpson for his invaluable assistance in deriving and expressing algorithms. This work was supported by a Curtin University Postgraduate Scholarship.

REFERENCES


In The International Symposium on Parallel and Distributed Computing, December 1993.

