

Embeddings into Hyper Petersen Networks: Yet Another Hypercube-Like Interconnection Topology*

SAJAL K. DAS, SABINE ÖHRING and AMIT K. BANERJEE†
Department of Computer Science, University of North Texas, Denton, TX 76203, USA

A new hypercube-like topology, called the *hyper Petersen* (HP) network, is proposed and analyzed, which is constructed from the well-known cartesian product of the binary hypercube and the Petersen graph of ten nodes.

This topology is an attractive candidate for multiprocessor interconnection having such desirable properties as regularity, high symmetry and connectivity, and logarithmic diameter. For example, an n -dimensional hyper Petersen network, HP_n , with $N = 1.25 * 2^n$ nodes is a regular graph of degree and node-connectivity n and diameter $n - 1$, whereas an $(n - 1)$ -dimensional binary hypercube, Q_{n-1} , with the same diameter covers only 2^{n-1} nodes, each of degree $(n - 1)$. Thus the HP topology accommodates 2.5 times extra nodes than Q_{n-1} at the cost of increasing the node-degree by one. With the same degree and connectivity of n , the diameter of the HP_n network is one less than that of Q_n , yet having 1.25 times larger number of nodes.

Efficient routing and broadcasting schemes are presented, and node-disjoint paths in HP_n are computed even under faulty conditions. The versatility of the hyper Petersen networks is emphasized by embedding rings, meshes, hypercubes and several tree-related topologies into it. Contrary to the hypercubes, rings of odd lengths, and a complete binary tree of height $n - 1$ permit subgraph embeddings in HP_n .

Key Words: *Communication, embedding, fault tolerance, hypercube, interconnection network, Petersen graph*

1 INTRODUCTION

One of the important aspects of a message-passing parallel computer is its interconnection network topology, which can be represented as an undirected graph such that each node represents a processor-memory module and an edge is a bidirectional communication link between two processors. Several network topologies have been proposed including trees, meshes, shuffle-exchange [1], hypercubes, cube connected cycles [2], pyramids [3], recursive networks [4], stars and pancakes [5], de Bruijn networks [6], folded Petersen networks [7] and so on.

Among these networks, the hypercubes have received significant attention because of their rich topological properties such as symmetry, high node-connectivity, logarithmic degree and diameter. Also, several commercial parallel machines (e.g. Intel

iPSC, NCUBE, Connection machine) use this topology to interconnect processors. In the recent years, various modifications have been incorporated into the hypercube structure to enhance its performance while preserving the useful properties. Twisted hypercube [8] and multiply twisted hypercube [9] provide two such examples, for which the diameter is reduced without increasing node degrees, whereas for folded hypercubes [10] and bridged hypercubes [11] the diameter is reduced with increasing node-degree.

Now, the fact that the well-known Petersen graph [12] of ten nodes has degree three and diameter two compared to a three-dimensional hypercube (with only eight nodes, each of degree three) having diameter equal to three, motivates us to design a new topology which makes use of the Petersen graph as a building block. At the same time our goal is to exploit the existing properties of the popular hypercube network. Thus a natural way to extend a hypercube is to replace each of its nodes by a Petersen graph, which is precisely the effect of a cartesian product. The proposed network constructed this way is called the *hyper Petersen* (HP) network, which

*This work was in part supported by Texas Advanced Research Program Grants under Award No. TARP 003594003 and TATP-003594031.

†Current address: National Airports Authority, Guwahati Airport, Guwahati, INDIA.

turns out to be an attractive alternative of the hypercubes for multiprocessor interconnection with slightly smaller diameter and higher packing density.

This paper also emphasizes the importance of the HP network as a versatile interconnection topology, by embedding various other structures into it. The guest networks include rings, grids, complete binary trees [13], full-ringed binary trees [14], meshes of trees [15], tree machines [16], X-trees [17], hypertrees [18], and pyramids [3]. These mapping problems are of practical importance. For example, an algorithm developed for any such guest network can be efficiently implemented onto the HP network, if there exists an efficient mapping of that guest into the HP host. Also if a problem has a tree-related task or data structure as in divide-and-conquer, branch-and-bound or image processing (quadtrees) algorithms, an efficient mapping of that structure onto the HP network is needed for an efficient parallel implementation of these algorithms on the HP.

The paper is organized as follows. In Section 2 we introduce various graph-theoretic terminology and definitions. Section 3 gives a formal definition of the HP network, while Section 4 discusses its topological properties. In Section 5 routing and broadcasting techniques are given. Section 6 shows how to embed other topologies into the HP network. Section 7 compares the embedding results with those in other host networks, and Section 8 concludes this paper.

2 DEFINITIONS AND TERMINOLOGY

An interconnection network is usually modeled as an undirected graph $G = (V, E)$ where V is the node-set representing processors and E the edge-set representing the communication links. We will use standard graph-theoretic terminology [12].

The *degree*, $\text{deg}(u)$, of a node u is the number of edges incident on u , and it is a measure of the network connection cost. If $\text{deg}(u) = \delta$, for $1 \leq u \leq |V| = N$, then G is called δ -regular, or a regular graph of degree δ . The *diameter* (d) of a network is a measure of its performance in terms of worst-case communication delay. It is defined as $d = \max\{\text{dist}_G(i, j) \mid 1 \leq i \leq j \leq N\}$, where $\text{dist}_G(i, j)$ is the shortest distance between nodes i and j in G . The *node-connectivity* (κ) is the number of nodes whose removal results in a disconnected network. It is a measure of fault-tolerance of the network. The f -fault diameter is defined as the worst-case diameter by removing at most f nodes. For a binary string $x = x_{k-1} \dots x_0$ we denote with $x(i) := x_{k-1} \dots x_{i+1} \bar{x}_i x_{i-1} \dots x_0$ the string x complemented in the i th bit, $0 \leq i \leq k-1$.

There are several graph-theoretic operations which enable us to aggregate large networks from smaller networks of known properties. One such operation is the so called *cartesian product* [12] of two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$. The resulting network $G = G_1 \times G_2 = (V, E)$ has the node-set $V = V_1 \times V_2$ and an edge $\{(x, y), (v, w)\} \in E$ iff either $x = v$ and $\{y, w\} \in E_2$ or $y = w$ and $\{x, v\} \in E_1$. Clearly, $G_1 \times G_2$ satisfies the following relations on the number of nodes, degree, diameter, and node-connectivity: $N = N_1 * N_2$, $\delta = \delta_1 + \delta_2$, $d = d_1 + d_2$, and $\kappa_1 + \kappa_2 \leq \kappa \leq \delta$.

An n -dimensional *binary hypercube*, Q_n , consisting of 2^n nodes has the node-set $V_n = \mathbf{Z}_2^n = \{x_{n-1} \dots x_0 \mid x_i \in \mathbf{Z}_2 \text{ for } 0 \leq i \leq n-1\}$, the set of binary strings of length n . Two nodes u and v are adjacent iff their *Hamming distance* is $H(u, v) = 1$, i.e. the binary labels of u and v differ in exactly one bit position. In terms of the cartesian product operation, hypercubes can be recursively defined as $Q_n = Q_{n-1} \times K_2$, where K_2 is the complete graph on two nodes. It is known that $\delta = d = \kappa = n$ for Q_n .

The *incomplete hypercube* [19] I_k^n , $0 \leq k < n$, has $2^n + 2^k$ nodes consisting of a *front cube* Q_n and a *back cube* Q_k . The node-set is given by the set of $(n+1)$ -bit labels such that (0^n) and $(1(0)^{n-k}*)$ denotes the front cube and the back cube, respectively, where $*$ is either 0 or 1. An edge between two nodes x and y of I_k^n exists iff x and y differ in exactly one bit.

The *undirected binary de Bruijn graph* [20] $DG(n)$ of order n is defined by the node-set $V_n = \mathbf{Z}_2^n = \{x_{n-1} \dots x_0 \mid x_i \in \mathbf{Z}_2 \text{ for } 0 \leq i \leq n-1\}$ and the edge-set $E_n = \{\{x_{n-1} \dots x_0, x_{n-2} \dots x_0 p\} \mid p, x_i \in \mathbf{Z}_2, 0 \leq i \leq n-1\}$. The order- (m, n) *hyper-de Bruijn graph* [21] is the product graph $HD(m, n) = Q_m \times DG(n)$.

An n -dimensional *twisted hypercube*, TQ_n , is constructed as follows. We select two distinct edges, say (u, v) and (w, z) , in a 4-cycle of the hypercube Q_n which have no nodes in common, and replace them with two new edges (u, z) and (v, w) [22]. Clearly, such a twist does not affect the degree of a node (i.e., $\delta = \kappa = n$), but the diameter of TQ_n is reduced to $d = n - 1$. A recursive application of the twist operation results in a *multiply twisted hypercube*, MTQ_n , having a significant reduction in the diameter (see Table 1).

3 HYPER PETERSEN NETWORKS

The *Petersen graph*, P , with ten nodes has an outer 5-cycle, an inner 5-cycle and five spokes joining them (see Figure 1a)). This is an important graph

TABLE I
Comparison of HP Network with Other Hypercube-like Networks

Network	HP_n	Q_n	TQ_n	MTQ_n	$DG(n)$	$HD(m, n)$	I_k^n
# nodes (N)	$1.25(2^n)$	2^n	2^n	2^n	2^n	2^{m+n}	$2^n + 2^k$
degree (δ)	n	n	n	n	4	$n + 4$	$k + 1 \leq \delta \leq n + 1$
diameter (d)	$n - 1$	n	$n - 1$	$\left\lceil \frac{n + 1}{2} \right\rceil$	n	$m + n$	$n + 1$
connectivity (κ)	n	n	n	n	2	$n + 2$	$k + 1$
cost ($C = d \cdot \delta$)	$n(n - 1)$	n^2	$n^2 - n$	$\frac{n(n + 1)}{2}$	$4n$	$(n + 4)(m + n)$	$(n + 1)^2$

belonging to the rare ‘‘Moore graphs’’ family, and has been studied extensively. For example, P is the smallest 3-regular graph in which the shortest cycle has length five. Therefore, it is also called ‘‘(3, 5)-cage.’’ Its diameter $d = 2$ and node-connectivity $\kappa = 3$. It is highly symmetric, but neither Hamiltonian nor planar. For more details on the Petersen graph, refer to [23].

There are several ways to characterize the node-adjacency of the Petersen graph. One way to construct this graph is by considering the ten subsets of size two (called *two-subsets*) of the set $\{1, 2, 3, 4, 5\}$ as nodes, and joining those pairs of subsets that are disjoint [23]. For example, there exists an edge between the pair of nodes $\{1, 2\}$ and $\{3, 4\}$ because they are disjoint subsets of size two. However, nodes $\{1, 2\}$ and $\{1, 3\}$ are not adjacent. To make the drawing less cumbersome, we designate a node $\{1, 2\}$ as 12 in Figure 1a). In the rest of the paper, we will instead use the decimal labeling of the nodes as also shown in Figure 1a).

The proposed n -dimensional (for $n \geq 3$) *hyper Petersen* network, $HP_n = Q_{n-3} \times P$, is a cartesian product of a $(n - 3)$ -dimensional hypercube and P . Informally, it is obtained from Q_{n-3} such that each of its nodes is replaced by a 10-node Petersen graph, P . For the sake of computational convenience, each node in HP_n is represented as two tuple: (u, v) , where u corresponds to an $(n - 3)$ -bit binary label of the

hypercube part, and v is a node in the Petersen graph. The label u is called the *prefix label* of a node (denoted either in binary or in decimal encoding in the following) and the corresponding Petersen graph will be denoted as P^u . Thus the basic building block of an HP_n is the 3-regular Petersen graph $P = HP_3$. Since $HP_n = Q_{n-4} \times K_2 \times P = HP_{n-1} \times K_2$, for $n \geq 4$, it can be recursively constructed by connecting two HP_{n-1} 's. We increase the number of bits in node labels of the first and second HP_{n-1} 's by one by adding a 0 and 1 in front of their prefix-labels, respectively. Next we connect together those nodes of these HP_{n-1} 's which differ only in the first bit (from left) of their prefix-labels. Figure 1b) depicts HP_4 .

One advantage of constructing a large network with the help of the cartesian product is that the characteristics of the component networks are preserved. For example, since both graphs Q_{n-3} and P are node- and edge-symmetric, HP_n is also node- and edge-symmetric.

4 PROPERTIES OF HP NETWORKS

An n -dimensional hyper Petersen network, HP_n , is a regular graph with $N = 2^n + 2^{n-2} = 1.25 * 2^n$ nodes, each of degree $\delta = n$, and having diameter $d = n - 1$. The prefix labels of any two nodes, say (x, u) and (y, v) , in HP_n can differ in at most $(n - 3)$ bits. Thus by complementing one bit at a time, the Petersen graph P^y to which (y, v) belongs can be reached from P^x in at most $n - 3$ steps. Then it requires at most two more steps to locate v within P^y .

Theorem 1: *The node-connectivity of HP_n is n . Its $(n - 1)$ -fault diameter is $n + 2$.*

Proof: The node-connectivity of HP_n is $\kappa = n$, since there are n node-disjoint paths between two arbitrary nodes (x, y) and (x', y') of length at most $dist_{HP_n}((x, y), (x', y')) + 3$ as shown below.

Let i_1, \dots, i_m be the m differing bits between x and x' , and let $k \in \{0, 1\}$, be the number of different

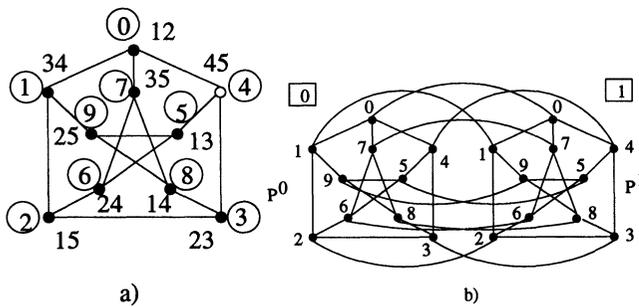


FIGURE 1 a) Petersen graph, P b) Hyper Petersen network $HP_4 = Q_1 \times P = HP_3 \times K_2$.

positions between y and y' such that $\alpha = \text{dist}_P(y, y')$. Furthermore, y^i , for $1 \leq i \leq 3$, are the neighbors of y in P . Let $\gamma = 1$ for $\alpha = 2$, otherwise $\gamma = 3$ for $\alpha = 1$, where γ is the difference between the lengths of a non-shortest and a shortest path in P connecting y and y' . Assume p_j , $j = 1, 2$, be the non-shortest paths in P between y and y' . Then there are $m + k$ node-disjoint shortest paths \mathcal{P}_j^1 , $1 \leq j \leq m + k$, (of length $r = m + k$) between (x, y) and (x', y') by equalizing the differing positions in cyclic order $(i_1, \dots, i_m, i_{m+k})$. Further $n - 3 - m$ paths of length $r + 2$ can be constructed for $j \notin \{i_1, \dots, i_m\}$ by the sequence $((x, y), (x(j), y))$, equalizing all elements except the j th element as in $\mathcal{P}_1^1, (x'(j), y'), (x', y')$, and $3(1 - k)$ paths of length $r + 2$ by the sequence $((x, y), (x, y^i))$, equalizing the bits in x and x' by using the path $\mathcal{P}_1^1, (x', y^i), (x', y')$. And $2k$ paths of length $r + \gamma$ can be constructed by the node-sequence $((x, y), (x, y^i))$, equalizing the bits i_1, \dots, i_m as in $\mathcal{P}_1^1, (x', y^i)$, changing the Petersen part labels as in path $p_j, (x', y')$. The node-disjointness of these constructed paths is proven by us in [24]. It follows that there exists still a path of length at most $\text{dist}_{HP_n}((x, y), (x', y')) + 3$ even after removal of $n - 1$ nodes. \square

Let us now compare the topological properties of HP_n with several other hypercube-like networks (Table 1). For the facts stated in this table, refer to [1, 25, 21, 19].

Maintaining the same degree $\delta = n$, the HP_n network can accommodate 1.25 times extra nodes than in Q_n , yet having the diameter reduced by one. On the other hand, considering diameter $d = (n - 1)$ as the criterion for comparison, HP_n has better connectivity and covers 2.5 times more nodes than the hypercube Q_{n-1} , only at the cost of increasing the node-degree by one. Although the degree and diameter of HP_n is identical to those of a twisted hypercube, TQ_n , the packing density is higher in the former network. As can be seen from Table 1, a special case of the incomplete hypercube, namely I_{n-2}^n , consisting of the same number of nodes as HP_n has larger diameter and poorer connectivity.

The de Bruijn graph $DG(n)$ provides a bounded degree network and has poor reliability, as compared to HP_n , which turns out to be a versatile network. It is worth pointing out that the de Bruijn graph is very efficient for (dynamic) embeddings of trees [26], but not for grids [27].

Contrary to the hyper-de Bruijn topology, HP_n provides higher symmetry and regularity. Also, the hyper Petersen network (HP_n) has reduced degree and diameter and higher connectivity than the hyper-

de Bruijn network $HD(n - 3, 3)$ consisting of 2^n nodes.

4.1 Fault Tolerance

Fault-tolerance capabilities of a network can be improved with the existence of a wide *container*, providing a large number of node-disjoint paths between any two nodes [10]. The *width* of a container is the number of node-disjoint paths it includes. The *length* of a container is the longest path length in it. The best container of a given width between two nodes is the one of the shortest length. The *container quality* between two nodes at a distance r in a node-symmetric network is given by $CQ(r) = (\Gamma_1/\Gamma_2)$, where Γ_1 is the number of node-disjoint paths and Γ_2 is the average length of all disjoint paths between those two nodes in the best container. A large value of $CQ(r)$ means a short container that includes many paths.

The container quality of HP_n (between two nodes (x, y) and (x', y') in HP_n at distance r) is given by $CQ_{HP_n}(r) = n^2/((m + k) \cdot r + (n - m - 3k) \cdot (r + 2) + 2k(r + \gamma)) = n^2/(n \cdot r + 2n - 2m - 6k + 2k\gamma)$. Here, m is again the number of different bits between x and x' and $k \in \{0, 1\}$ the number of different positions between y and y' . If $k = 0$ and thus $r = m$, then $CQ_{HP_n} = CQ_{Q_n} = n^2/(n \cdot (r + 2) - 2r)$ [10]. On the other hand, if $k = 1$, then $CQ_{HP_n} = n^2/(n \cdot r + 2n - 2m - 6 + 2\gamma) = n^2/(n(r + 2) - 2(m + 2))$ if $\alpha = 2$, otherwise $CQ_{HP_n} = n^2/(n \cdot r + 2n - 2m)$, if $\alpha = 1$. Thus, $(CQ_{HP_n}/CQ_{Q_n}) = 1$ for $\alpha = 2$ and $(CQ_{HP_n}/CQ_{Q_n}) = (nr + 2n - 2m - 2)/(nr + 2n - 2m) < 1$ for $\alpha = 1$. This means that HP_n provides a container that is approximately as good as in Q_n .

Let us now compute the probability of failure of HP_n . Since its node-connectivity, upto $(n - 1)$ node faults can be tolerated without disconnecting the network. Also, HP_n will be disconnected if all the neighboring processors of a particular node are faulty. Let us denote the probability of occurrence of such a faulty event as $F(HP_n)$, and let the corresponding parameter for a binary hypercube, Q_n , be $F(Q_n)$. Suppose the binomial coefficient, $C(i, j) = \binom{i}{j}$, denotes the number of ways of choosing j objects out of i objects.

Then considering a fault set of n nodes in a Q_n , there exists $C(2^n, n)$ different faults. Among these only 2^n faults result in disconnecting the network, and again each such set consists of nothing but the set of all neighbors of a given node. Similar logic applies to a hyper Petersen topology. Thus, $F(Q_n) =$

$(2^n/C(2^n, n))$, and $F(HP_n) = (1.25 * 2^n)/(C(1.25 * 2^n, n))$ such that $(F(HP_n)/F(Q_n)) = (1.25 * C(2^n, n))/C(1.25 * 2^n, n) \ll 1$. It implies that for a given pattern of faulty nodes, the probability of disruption of an n -dimensional hyper Petersen network is significantly less than that of Q_n . In other words, on the average the fault tolerance capability of HP_n is better.

4.2 Routing and Broadcasting

For routing, let $S = (x, y)$ and $D = (x', y')$ be the source and destination nodes, respectively, where $S \neq D$. If $x = x'$, the routing is restricted within a single Petersen graph, P^x , from node y to y' and requires at most two hops due to the diameter of the Petersen graph. When $x \neq x'$, we equalize the differing bit-positions between x and x' in a sequential order until we reach the node (x', y) . If $y = y'$, accept the message; otherwise route it within the Petersen graph $P^{x'}$ until the destination is found. This procedure ensures that point-to-point message routing in HP_n is via a shortest path and requires at most $(n - 1)$ hops.

In the case of at most $n - 1$ faults, and when each node knows the status of each other node in HP_n , the routing is done along one of the n node-disjoint paths of length at most $dist_{HP_n}(S, D) + 3$ (cf. Theorem 1). On the other hand, if each node (u, v) has only local knowledge about the status of $2^{n-3} + 8$ other nodes (having the hypercube-part label u or the Petersen-part label v), a path of length $\leq n + 7$ can be found. First try to send the message to the intermediate node (x, y') using routing in P for $y \neq y'$. If this is not possible, since (x, y') or at least three nodes in the subgraph P^x are faulty, try to route the message from (x, y) to (x', y) using hypercube routing, if $x \neq x'$. If this is not possible due to (x', y) or at least $n - 3$ nodes in Q_{n-3}^y being faulty, request a path from the hypercube part neighbors $(x(i), y)$, $0 \leq i \leq n - 4$, and the Petersen-part neighbors (x, y^i) , $0 \leq i \leq 2$. Because of at most $n - 1$ faulty nodes, one of these paths is operational and its length is at most $n + 7$.

The broadcasting commonly used in a hypercube can be easily adopted for the hyper Petersen configuration. In Q_n , the nodes identify the communication channels according to their dimensions, i.e., channel i for the neighbor at dimension i , for $0 \leq i \leq n - 1$. Broadcasting in Q_n requires n hops. Now, in HP_n , broadcasting will require a total of $(n - 1)$ hops because within a Petersen graph only two hops are required to broadcast to all 10 nodes, and $(n - 3)$ hops are sufficient to broadcast to all nodes in 2^{n-3}

Petersen graphs. This implies that in the HP_n network, we can broadcast a message to 2.5 times additional nodes than in Q_{n-1} with the same number of hops. A broadcast tree is depicted in Figure 2. The broadcasting scheme suggests us to calculate the average distance in HP_n .

Theorem 2: *The number Z_i^n of nodes at distance i from any node in HP_n is $Z_i^n = \binom{n-3}{i} + 3 \cdot \binom{n-3}{i-1} + \frac{6}{i-2} \cdot \binom{n-3}{i-2}$. The average distance in HP_n is given by $\bar{d}_n = ((2^n + 2^{n-2}) \cdot n)/((2^n + 2^{n-2} - 1) \cdot 2) \approx (n/2)$.*

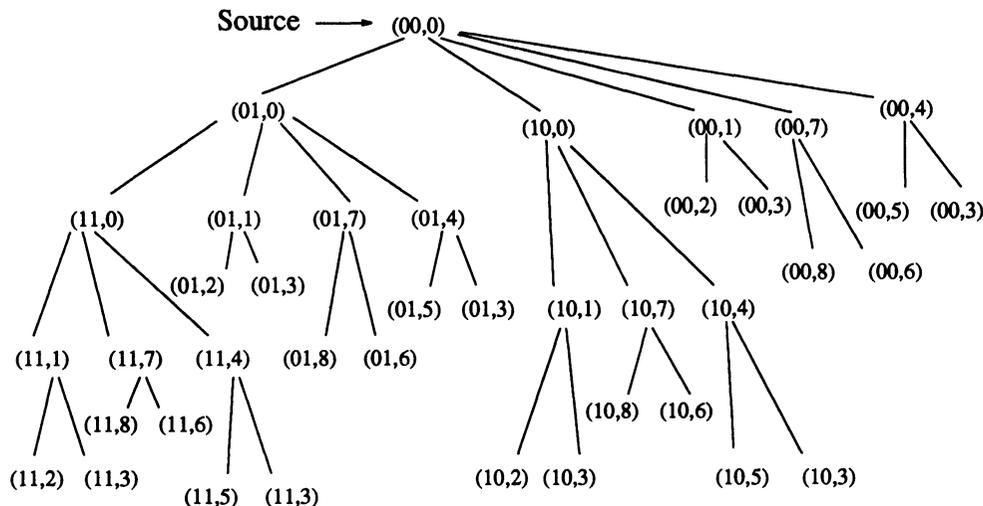
Proof: HP_n is node-symmetric. Thus, w.l.o.g., let us compute the number of nodes Z_i^n at distance i , where $1 \leq i \leq n - 1$, from the node $((0)^{n-3}, 0)$. It is obvious that $Z_i^n = \tilde{Z}_i^{n-3} + 3 \cdot \tilde{Z}_{i-1}^{n-3} + 6 \cdot \tilde{Z}_{i-2}^{n-3}$, where $\tilde{Z}_i^n = \binom{n}{i}$ is the number of nodes of distance i from any node in the hypercube Q_n . Thus, $\tilde{Z}_i^n = \binom{n-3}{i} + 3 \cdot \binom{n-3}{i-1} + 6 \cdot \binom{n-3}{i-2}$. The average distance $\bar{d}_n = (\sum_{i=1}^{n-1} i \cdot [(\binom{n-3}{i} + 3 \cdot \binom{n-3}{i-1} + 6 \cdot \binom{n-3}{i-2})]) / (2^n + 2^{n-2} - 1) = ((2^n + 2^{n-2}) \cdot n) / (2^n + 2^{n-2} - 1) \approx (n/2)$, which is almost the same as in the binary hypercube, Q_n . \square

5 EMBEDDINGS INTO HP NETWORKS

This section deals with the mapping of several standard topologies into the HP network. An *embedding* of a *guest* graph $G = (V_G, E_G)$ into a *host* graph $H = (V_H, E_H)$ is a one-to-one mapping of their vertices along with a mapping $\psi: E_G \rightarrow \{\text{paths in } H\}$, from edges in G to paths in H . The *dilation* of an edge, $e \in E_G$, is the length of path $\psi(e)$ in H . The dilation of an embedding is the maximum dilation over all edges. The *expansion* of an embedding is $|V_H|/|V_G|$. An embedding with expansion of one is said to be an embedding of a graph into the *optimal sized* host [28]. The *edge-congestion* is the maximum number of edges of G that are routed by g over a single edge of H . An embedding is *modular*, if the embedding of a guest graph G of size or dimension $n + 1$ into the optimum-sized host is an extension of the embedding of G of size n into its optimum-sized host.

5.1 Rings and Linear Arrays

Although the Petersen graph does not contain a Hamiltonian cycle (one which contains all nodes exactly once), it is easy to show that the hyper Petersen network HP_n , for $n \geq 4$, is Hamiltonian. Furthermore, unlike the binary hypercubes, rings of odd

FIGURE 2 A broadcast tree in $HP_5 = Q_2 \times P$.

lengths can be embedded into hyper Petersen networks and this property increases its flexibility.

Theorem 3: Any ring (hence linear array) of l nodes, $4 \leq l \leq 1.25(2^n)$, is a subgraph of HP_n for $n \geq 4$.

Proof (by induction): First, we prove that in HP_4 one can find a ring R_l of length l , $4 \leq l \leq 20$, containing the edge $\{(0, 0), (0, 1)\}$. In the Petersen graph P , there exists a linear array of length i , $1 \leq i \leq 10$, namely $(0, 1, 2, \dots, 9)$. Thus the sequence of nodes $(0, 0), \dots, (0, l-1), (1, l-1), \dots, (1, 0)$ builds a ring of length $2l$ in HP_4 , where $2 \leq l \leq 10$. Furthermore, one can easily prove that each ring R_l , where l is odd and $5 \leq l \leq 19$, can be embedded as subgraph in HP_4 such that R_l contains the edge $\{(0, 0), (0, 1)\}$.

Now, as an induction hypothesis assume that in HP_n each ring of length l , for $4 \leq l \leq 2^n + 2^{n-2}$, can be embedded as subgraph such that it contains the edge $\{(x, 0), (x, 1)\}$ for some node $x \in Q_{n-3}$. Since each ring R_l in HP_n is also contained in HP_{n+1} , we only have to prove the existence of R_l in HP_{n+1} with an edge $\{(x, 0), (x, 1)\}$, $x \in Q_{n-2}$, for $2^n + 2^{n-2} + 1 \leq l \leq 2^{n+1} + 2^{n-1}$. Thus, $l = l_1 + l_2$ with $4 \leq l_1, l_2 \leq 2^n + 2^{n-2}$. Now, consider the rings R_{l_1} embedded in $0HP_n$ and R_{l_2} in $1HP_n$. According to the inductive assumption, R_{l_1} has an edge $\{(0x, 0), (0x, 1)\}$, while R_{l_2} has an edge $\{(1x', 0), (1x', 1)\}$ for some $x' \in Q_{n-3}$. Since the hypercube is node-symmetric, we use a mapping f_1 on $0Q_{n-3}$ that maps the node $0x$ to the node $0(0)^{n-3}$ and a mapping f_2 on $1Q_{n-3}$ that maps the node $1x'$ on $1(0)^{n-3}$. Next, we divide the rings R_{l_1} and R_{l_2} at the edges $\{(0(0)^{n-3}, 0), (0(0)^{n-3}, 1)\}$ and $\{(1(0)^{n-3}, 0), (1(0)^{n-3}, 1)\}$, respec-

tively, and connect the two rings as shown in Figure 3a).

To guarantee that the obtained ring contains an edge $\{(y, 0), (y, 1)\}$, for some $y \in \{0, 1\}$, we use an automorphism of P that maps an arbitrary edge $\{(y, i), (y, j)\}$ to the edge $\{(y, 0), (y, 1)\}$. This is always possible, since the Petersen graph P is edge-symmetric. \square

5.2 Hypercube Embedding

Theorem 4: A Hypercube Q_n and additionally a Q_{n-2} can be embedded in a node- and edge-disjoint manner in HP_n with load 1, dilation and edge-congestion 2. There also exists an optimal embedding of the incomplete hypercube I_{n-2}^* into the optimum sized host HP_n with load 1 and dilation and edge-congestion 2.

Proof: Figure 3b) illustrates the embedding of Q_3 in the Petersen graph P .

Then we continue with two copies of Q_3 embedded in P^0 and P^1 , respectively, and connect them by combining the corresponding nodes in Q_4 . Assume Q_{n-1} is embedded in HP_{n-1} . Then two copies of Q_{n-1} 's embedded in $0HP_{n-1}$ and $1HP_{n-1}$ are combined through the corresponding nodes to yield Q_n in HP_n . Since the nodes $(x, 0)$ and $(x, 8)$, where $x \in \mathbb{Z}_2^{n-3}$, are not used so far, they can now be utilized to embed an additional Q_{n-2} in a node- and edge-disjoint fashion with load 1, dilation and edge-congestion 2.

To show the embeddings of the incomplete hypercubes, consider the embedding (f_3, g_3) of I_1^* in P as in Figure 3c). Here g_3 is indicated with dashed

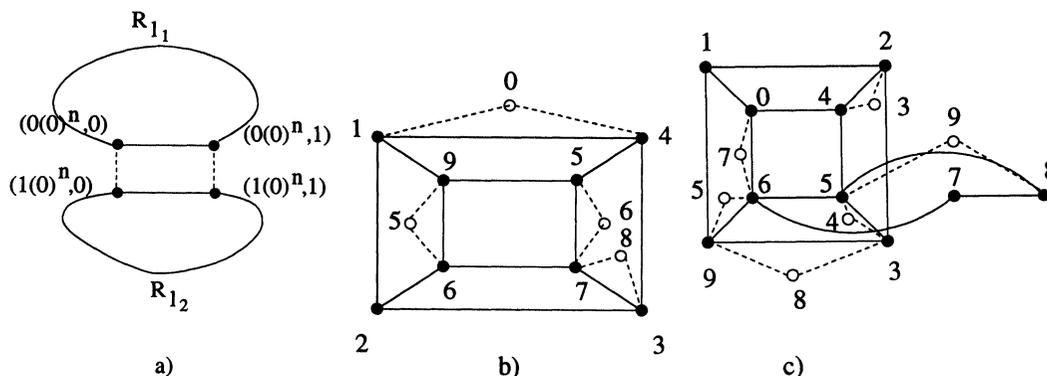


FIGURE 3 a) Connection of R_{11} and R_{12} b) Embedding Q_3 in P c) Embedding I_1^1 in P .

lines and white nodes, as far as the edge is not mapped to an edge in the host. The nodes v in I_1^1 are labelled with their image-nodes $f_3(v)$ in P . Now, $I_{n-2}^n = I_1^1 \times Q_{n-3}$ can be embedded in $HP_n = Q_{n-3} \times P$ using the node-mapping $\tilde{f}_n(xy) = (y, f_3(x))$, where $y \in \mathbb{Z}_2^{n-3}$ and $x \in \{0(*)^3, 100*\}$, and (f_3, g_3) is the embedding defined above. We can additionally prove that this embedding is optimal, that is I_{n-2}^n cannot be embedded with adjacency preserving in HP_n . Both HP_n and I_{n-2}^n have equal-sized node- and edge-set. Thus, if I_{n-2}^n is a subgraph of HP_n , the two graphs would be isomorphic. This is a contradiction, since I_{n-2}^n is bipartite, but HP_n is not. \square

5.3 Mesh Embedding

Theorem 5: A $2^i \times 2^j \cdot 10$ mesh, for $0 \leq i + j \leq n - 3$, is a subgraph of HP_n . A $2^{n-2} \times 5$ mesh (or $3 \cdot 2^i \times 3 \cdot 2^j$ mesh), for $i + j = n - 3$, can be embedded in HP_n with dilation and edge-congestion 2 and expansion 1 (or 10/9).

Proof: We use the k -bit binary reflected Gray codes defined recursively as $\mathcal{G}_1 = \{0, 1\}$, $\mathcal{G}_k = \{g_0^k, \dots, g_{2^k-1}^k\}$, and $\mathcal{G}_{k+1} = \{0g_0^k, 1g_0^k, 1g_1^k, 0g_1^k, \dots, 1g_{2^k-1}^k, 0g_{2^k-1}^k\}$. A $2^i \times 2^j \cdot 10$ mesh can be embedded as subgraph in the HP_n network using the binary reflected Gray codes \mathcal{G}_i and \mathcal{G}_j and the fact that a linear array of length 10 is a subgraph of P .

Figures 4a) and b) illustrate a 3×3 mesh and a 2×5 mesh obtained from the Petersen graph. Although it is an optimal sized embedding, the dilation is two. Figure 4a) and b) form the basic building block for our construction of meshes in higher dimensional hyper Petersen networks. Using the embedding of a 2×5 mesh in P and the binary reflected Gray code \mathcal{G}_{n-3} , a $2^{n-2} \times 5$ mesh can be embedded in HP_n with expansion 1 and dilation and edge-congestion 2. The embedding of a 3×3 mesh

in P together with the binary reflected Gray code \mathcal{G}_i and \mathcal{G}_j leads to an embedding of a $3 \cdot 2^i \times 3 \cdot 2^j$ mesh in HP_n with expansion 10/9 and dilation and edge-congestion 2.

For illustration, let us construct a 6×6 mesh (see Figure 4c)) in the HP_5 network. We first embed 3×3 meshes in Petersen graphs P^{00} and P^{01} which differ by one bit in dimension 0, and then take vertical mirror reflection (we call it *vertical swap*) of the mesh formed in P^{01} . Now join nodes along the leftmost column of the mesh in P^{01} with the corresponding nodes along the rightmost column of the mesh contained in P^{00} . This procedure forms a mesh of order 3×6 . Again considering the Petersen graphs P^{10} and P^{11} , we similarly construct a 3×6 mesh. This mesh is then swapped horizontally such that the nodes along the bottom row of the meshes obtained from P^{00} and P^{01} are now directly above their corresponding nodes along the uppermost row of the meshes obtained from P^{10} and P^{11} , respectively. This yields a mesh of order 6×6 . \square

The following subsections describe the embeddings of tree-related structures into the HP topology. The guest networks include complete binary trees, X-trees, full-ringed binary trees, tree-machines, hypertrees and pyramids into the HP topology.

5.4 Complete Binary Trees

Complete binary trees are useful computational structures both for broadcasting and applications in dictionary and database operations [14].

A *complete binary tree* $CBT(n) = (V_n, E_n)$ of height n is defined by $V_n = \{x_j \dots x_0 \mid x_k \in \mathbb{Z}_2, 0 \leq k \leq j, 0 \leq j \leq n - 1\} \cup \{\epsilon\}$ and $E_n = \{\{x, xa\} \mid x \in \mathbb{Z}_2^j, a \in \mathbb{Z}_2, 0 \leq j \leq n - 1\}$, where ϵ is the empty string. The node ϵ is called the *root* of the tree at

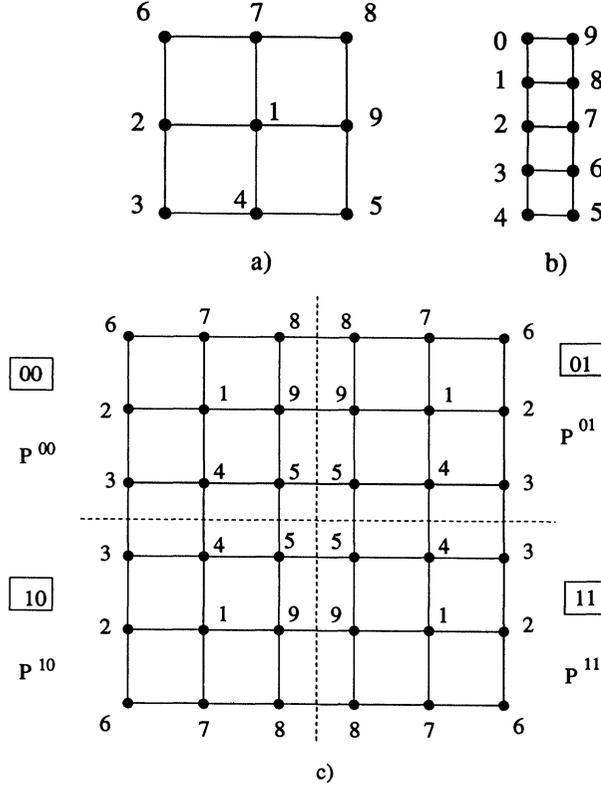


FIGURE 4 a) A 3×3 mesh in P b) A 2×5 mesh in P c) A 6×6 mesh in HP_5 .

level 0, and a node $x = x_i \dots x_0$ is at level $i + 1$. Let $MetaCBT(n)$ denote the tree $CBT(n)$ with an additional root, the so-called *metaroot* (cf. Figure 5a)). In this subsection, an optimum sized complete binary tree $CBT(n)$ is embedded as a subgraph in HP_{n+1} .

Theorem 6: *A complete binary tree $CBT(n)$ of height n is a subgraph of HP_{n+1} .*

Proof: Figure 5a) shows that $MetaCBT(2)$ is a subgraph of the Petersen graph $P^x, \forall x \in Q_{n-2}$. Due to the symmetry and degree of P^x , for each edge $\{v, w\}$ of the Petersen graph, there exists exactly one $MetaCBT(2)$ in P^x with (x, v) as the metaroot and (x, w) as the root, if we additionally demand that each left child must have a smaller suffix-label than the corresponding right child.

Let $M_{x,y}^{x,z}$ denote the embedded $MetaCBT(2)$ with root (x, y) and metaroot (x, z) . Let $M_{x,y}^{(x,z)}$ the $MetaCBT(2)$ having root-node (x, y) but without the metaroot. In the following embedding, $pred(i) = i - 1$ for $i \geq 1$ and $pred(0) = 4$. The tree starts modularly with the lower left subtree of height two, as depicted in Figure 5b). Here level l of $CBT(n)$ corresponds to the row number $r = n - l - 3$. For each current subtree to be embedded, we calculate in HP_{n+1} the root and the metaroot which is the parent

node of the next higher subtree with the current subtree as left subtree (cf. Figure 5b)). We start with $r = -1$ at the left lowest subtree of height two, by using the embedded $MetaCBT(2)$ in Figure 5c), i.e., $M_{0,1}^{(0,0)}$ with root $(0, 1)$ but without metaroot $(0, 0)$.

In row $r = 0$, the lower left subtree is constructed, such that $(0, 4)$ is the metaroot of the new root $(0, 0)$.

For $r = \text{odd}$, the new root is mapped to the metaroot node (x, i) of the left subtree while the new metaroot is mapped on $(x + 2^{(r-2)^+}, i)$, where $x_+ = \max\{0, x\}$. The right subtree of the root (x, i) is embedded by taking a copy of the left subtree, adding the offset-value 2^r to the prefix-label of each node (y, j) , and choosing $pred(j)$ as the suffix-label to guarantee distance one between the current node and its right son.

If row $r = \text{even}$, we choose as new root node again the metaroot of the left subtree. Consider the root $\tilde{r} = (y, \tilde{i})$ of the left subtree rooted in row $r - 2$. Due to the embedding algorithm, x and y differ in exactly the $(r - 3)_+$ th bit and $i = pred(\tilde{i})$. Let (y, i) be the left child of the root (x, i) , and let $(x, pred(i))$ be chosen as the new metaroot. Now we take a copy of the left subtree and add the offset 2^r to the prefix-labels. In order to guarantee load and dilation of 1, we exchange the suffix-labels for the left and right children (and their corresponding metaroots) starting at row $(r - 3)_+$ down to the leaves. For rows $(r - 3)_+$ upto $r - 1$, we add the offset $2^{(r-3)^+}$ to the prefix-label of each node and its metaroot. As an example, an embedding of $CBT(5)$ is shown in Figure 5c).

5.5 X-trees and Full-Ringed Binary Trees

An X -tree of height n , denoted by $X(n)$, consists of a complete binary tree $CBT(n)$ of height n with additional edges connecting the vertices at the same level [17].

The network $X(n)$ is a subgraph of the *full-ringed binary tree* of height n , denoted by $FRBT(n)$, which is a $CBT(n)$ such that all nodes at each level are connected as a cycle. Figure 6 depicts $X(3)$ and $FRBT(3)$.

Theorem 7: *There exists an optimal modular embedding $(\tilde{f}_n, \tilde{g}_n)$ of an X -tree (and full-ringed binary tree) of height $n - 1$ into HP_n with load 1, dilation and edge-congestion 2.*

Proof: The following construction defines the node mapping \tilde{f}_n . An X -tree of height 2 with an additional metaroot \tilde{r} can be embedded with load 1, edge-congestion 2, and dilation 2 into P^0 (cf. Figure 7a)). This

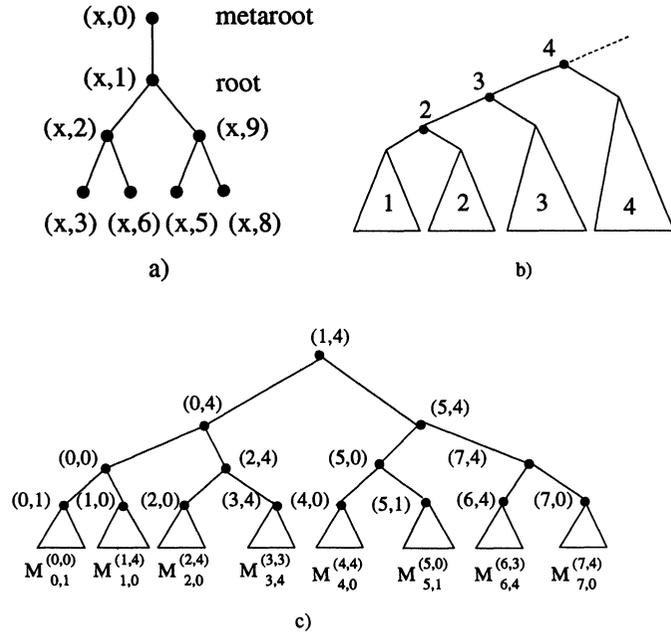


FIGURE 5 a) $MetaCBT(2)$, $M_{x,2}^{\alpha}$ b) Modular Construction c) Embedding of $CBT(5)$.

holds also for $FRBT(2)$, since $\{2, 3\}$ is an edge in the Petersen graph P . The dashed edge $\{x, y\}$ in Figure 7a) means that the edge between the two nodes x and y is mapped to a path of length 2 in P , where the intermediate node is indicated by an arrow. Thereby P^0 denotes the Petersen graph with prefix-label 0 in HP_n .

When the embedding of $X(2 + i + 1)$, for $0 \leq i \leq n - 4$, is constructed, one instance of an embedded $X(2 + i)$ is connected with the mirror image (by vertical reflection) of another instance of an embedded $X(2 + i)$, where an offset-value of $\alpha_i = 2^i$ is added to the prefix-labels. This construction is depicted in Figure 8a), where $m(G)$ denotes the mirror image of the graph G by vertical reflection and $f'(X(2 + i))$ is the embedding of $X(2 + i)$, where the offset 2^i is added to the prefix-labels. The root of $X(2 + i + 1)$ is placed at the position of the metaroot of $f(X(2 + i))$ and the metaroot of $X(2 + i)$ is placed at the position of the metaroot of $m(f'(X(2 + i)))$.

This implies for any two nodes $v_1, v_2 \in X(2 + i$

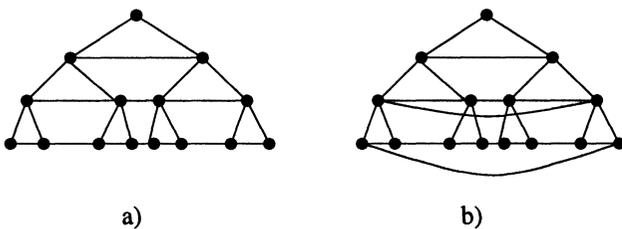


FIGURE 6 a) $X(3)$ and b) $FRBT(3)$.

+ 1) with level $(v_1) = \text{level}(v_2) = l$ and v_1 (or v_2) is the leftmost (or rightmost) node in level l , that the edge $\{v_1, v_2\}$ is mapped to an edge in HP_n .

The edge routing \tilde{g}_n is given as:

$$\tilde{g}_n(\{x, y\}) := \begin{cases} \{\tilde{f}_n(x), \tilde{f}_n(y)\} & \text{if } \text{dist}_{HP_n}(\tilde{f}_n(x), \tilde{f}_n(y)) = 1 \\ (\tilde{f}_n(x), \tilde{f}_n(\text{parent}(x)), \tilde{f}_n(y)) & \text{if } \tilde{f}_n(x), \tilde{f}_n(y) \text{ differ only in the suffix-label} \\ (\tilde{f}_n(x), (\Pi_1^2(\tilde{f}_n(y)), \Pi_2^2(\tilde{f}_n(x))), \tilde{f}_n(y)) & \text{if } \tilde{f}_n(x), \tilde{f}_n(y) \text{ differ in the prefix- and the suffix-labels.} \\ (\tilde{f}_n(x), (\Pi_1^2(\tilde{f}_n(x))(i + 1), \Pi_2^2(\tilde{f}_n(x))), \tilde{f}_n(y)) & \text{if } \tilde{f}_n(x), \tilde{f}_n(y) \text{ differ in bit } i \text{ and } i + 1 \text{ of the prefix-label.} \end{cases}$$

Thereby, $\Pi_i^k : \mathbb{N}^k \rightarrow \mathbb{N}$, for $1 \leq i \leq k$, denotes the i th projection of \mathbb{N}^k such that $\Pi_i^k(n_1, \dots, n_i, \dots, n_k) = n_i$. An example of this embedding is depicted in Figure 7b). There, each subtree of height two denoted by a triangle with label i stands for the embedding of a $CBT(2)$ in P^i as in Figure 7a).

It is easy to see that the embedding is optimal. Let us assume on the contrary that an $X(h)$, for $h \geq 1$, is a subgraph of HP_n . Then HP_n must contain at least one cycle of length 3 as subgraph, which is a contradiction. \square

Remark: \tilde{f}_n does not use the host nodes $\{(x, j) \mid x \in \mathbb{Z}_2^{n-3}, j \in \{1, 4\}\} \cup \{((1)^{n-3}, 0)\}$. These $2^{n-2} + 1$

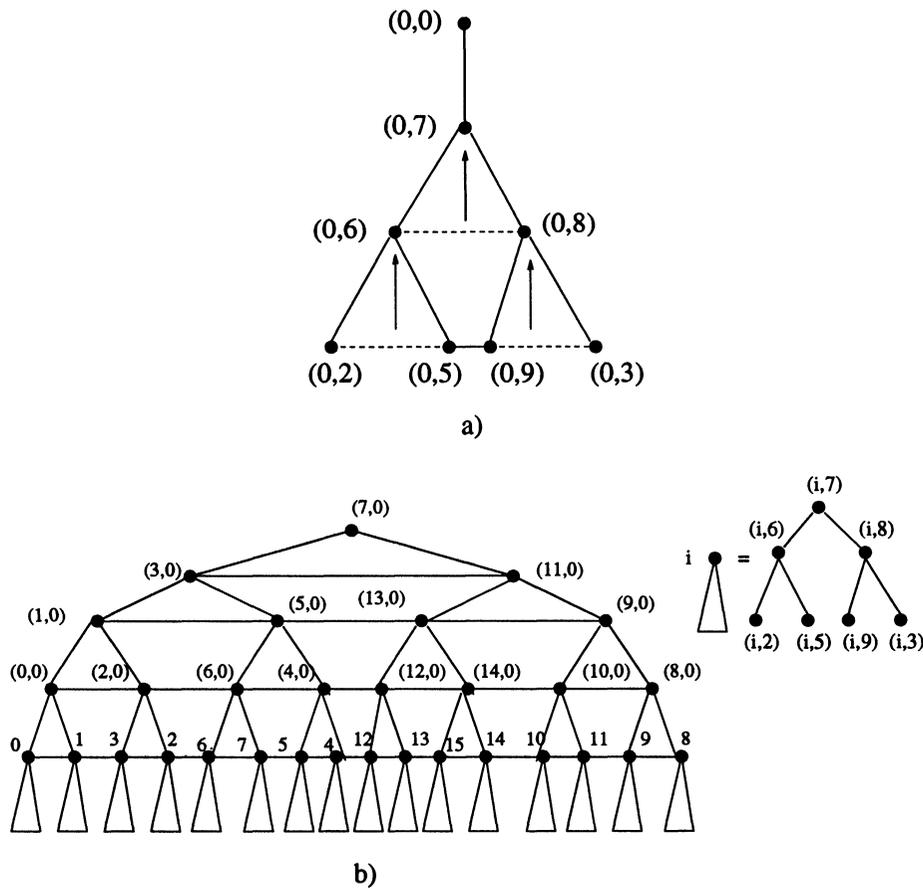


FIGURE 7 a) Embedding $X(2)$ with metaroot in P^0 b) Scheme of embedding $X(6)$.

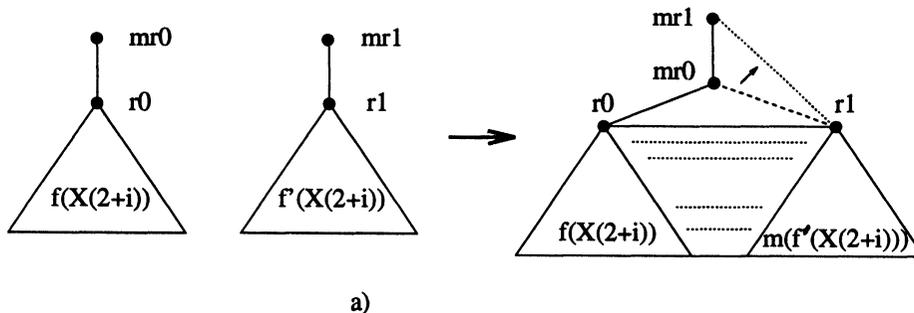
nodes of HP_n can be used to embed another $X(n - 3)$ which is node- and edge-disjoint from $\tilde{f}_n(X(n - 1))$. An example is illustrated in Figure 8b).

5.6 Hypertrees

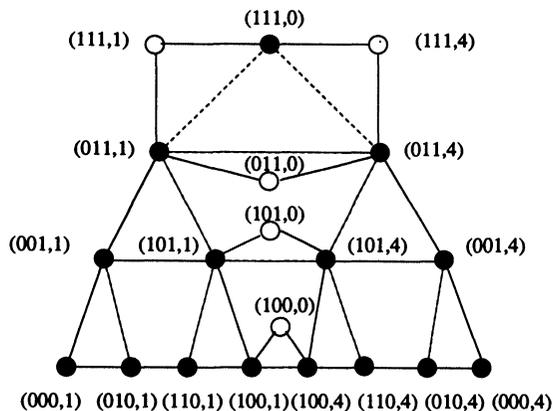
A *hypertree*, $HT(n)$, of height n is a complete binary tree $CBT(n)$ with additional links between the nodes at each level of the tree [18]. Assume that the nodes are numbered in breadth-first search manner (in binary encoding) as in Figure 9a). Let $r = z + 1$, where z denotes the number of consecutive trailing zeros in the binary encoded level number. The specific bit number b which is affected by the hypercube connections at level m can be determined by $b = (m/2^z) + 1/2$. Thereby, the bits are counted from the left, such that the most significant bit in the address is counted as bit 0. Figure 9a) shows a hypertree $HT(3)$.

Theorem 8: *There exists an optimal modular embedding of a hypertree $HT(n)$ of height n into HP_{n+1} with load 1, dilation 2 and edge-congestion 2.*

Proof: We prove this theorem by induction on the height h of the tree $HT(h)$. Let $\tilde{HT}(h)$ be the graph consisting of a $CBT(h)$ with additional links between the nodes x numbered by the bitstrings (of length $i + 1$) of 2^i to $2^{i+1} - 1$ in level i , $0 \leq i \leq h$. Each node x in level i is connected with all nodes $x(k)$, $0 \leq k < \lfloor \log_2 i \rfloor$ in level i . Clearly, $HT(h)$ is a subgraph of $\tilde{HT}(h)$. We even prove that the graph $\tilde{HT}(n)$ can be embedded in HP_{n+1} with the claimed cost measures. For a modular embedding (in each step of which an entire level of leaves is inserted), such a hypertree makes only sense when it grows from the root to the leaves. Accordingly, the mapping starts with the root of $\tilde{HT}(h)$ mapped to $(2^{n-3} - 1, 0)$. For level $l = 0$ to $n - 4$, the left (or right) son of an already embedded node v in level l with $f(v) = (x, 0)$ is mapped to the node $(x - 2^{n-4-l}, 0)$ or $(x +$

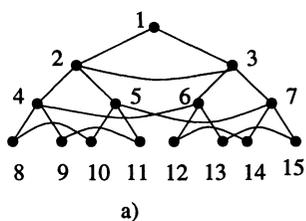


a)

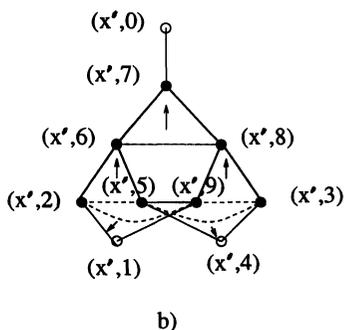


b)

FIGURE 8 a) Embedding of $X(2 + i + 1)$ b) Embedding of an additional $X(3)$ into HP_6 .



a)



b)

FIGURE 9 a) Hypertree of height 3 b) Embedding of $HT(2)$ into P^x .

$2^{n-4-l}, 0)$, respectively. In level $l = n - 3$, the node v mapped to $(x, 0)$ has as left subtree the tree $\hat{HT}(2)$ embedded into P^x depicted in Figure 9b) (without the additional root) and as right subtree the tree $\hat{HT}(2)$ embedded into P^{x+1} . \square

5.7 Tree Machines

A tree machine of dimension n , denoted as $TM(n)$, consists of two $CBT(n)$'s connected back to back along the common leaves. The dark lines in Figure 10a) depict $TM(2)$.

Theorem 9: *There exists a modular embedding of $TM(n)$ into HP_{n+2} for all $n \geq 3$ (into HP_3 for $n \leq 2$) with load 1, dilation 2, edge-congestion 2 and expansion $1 \leq (2^{n+2} + 2^n)/(2^{n+1} + 2^n - 2) \leq 2$.*

Proof: For the embedding of $TM(2)$ into HP_3 the mapping in Figure 10a) is used, where the edge between $(0, 5)$ and $(0, 1)$ is routed over node $(0, 9)$ and

the edge between (0, 9) and (0, 4) is routed over (0, 5).

For $2 \leq i \leq n - 1$, we take two instances of the embedded $TM(i)$, add the off-set 2^{i-1} to the prefix-labels of the second instance, and then join them inserting two roots $(2^{i-1} - 1, 7)$ and $(2^{i-1} - 1, 0)$ according to the decimal encoding of the prefix-labels (cf. Figure 10b). The preceding algorithm describes the node mapping f . Let (f_n, g_n) be the embedding of a $CBT(n)$ in Q_{n+1} by inorder numbering of the vertices. For the edge routing g , we define

$$g(\{x, y\}) = \begin{cases} \{f(x), f(y)\} & \text{if } \text{dist}_{HP_{n+2}}(f(x), f(y)) = 1 \dots (1) \\ \bar{g}'_n(\{x, y\}, \Pi_2^2(f(x))) & \text{if not } (1), \Pi_2^2(f(x)) = \Pi_2^2(f(y)) \\ ((\Pi_1^2(f(x)), 5), (\Pi_1^2(f(x)), 9), (\Pi_1^2(f(y)), 1)) & \text{if not } (1), \Pi_2^2(f(x)) = 5, \Pi_2^2(f(y)) = 1 \\ ((\Pi_1^2(f(x)), 9), (\Pi_1^2(f(x)), 5), (\Pi_1^2(f(y)), 4)) & \text{if not } (1), \Pi_2^2(f(x)) = 9, \Pi_2^2(f(y)) = 4 \end{cases}$$

Thereby, $\bar{g}'_n(e, x) = ((v_1, x), (v_2, x), (v_3, x))$, with $g_{n-2}(e) = (v_1, v_2, v_3)$.

Remarks:

- (1) There remain $2^{n+1} + 2$ nodes of HP_{n+2} which are not used by the embedding just discussed. These nodes are $\{(2^{n-1} - 1, 0), (2^{n-1} - 1, 7)\} \cup \{(x, j) \mid x \in \mathbb{Z}_2^{n-1} \setminus \{y0 \mid y \in \mathbb{Z}_2^{n-2}\}, j \in \{1, 2, 3, 4, 5, 6, 8, 9\}\}$, and they can be used for embedding either an additional $TM(n - 1)$ or another complete binary tree $CBT(n)$. For details, refer to [29].
- (2) For $k = 2, 3$ we can show that $TM(k)$ is a subgraph of HP_{k+2} . The question whether this approach can be generalized for $k > 2$ is still open.

5.8 Mesh of Trees

The two-dimensional $2^n \times 2^n$ mesh of trees [13], denoted by $MT(n, n)$, is the graph whose nodes are all pairs (x, y) , where x and y are binary strings of length at most n and at least one of them has length exactly n . The edges in $MT(n, n)$ connect (x, y) with (xa, y) , where $a \in \{0, 1\}$, if the length of x is less than n , and (x, y) with (x, ya) , if the length of y is less than n . Figure 11 illustrates $MT(2, 2)$.

The network $MT(m, n)$ is obtained from the $2^m \times 2^n$ mesh by eliminating all mesh edges, erecting a copy of the complete binary tree $CBT(m)$ along each column, using the column nodes as leaves of the tree,

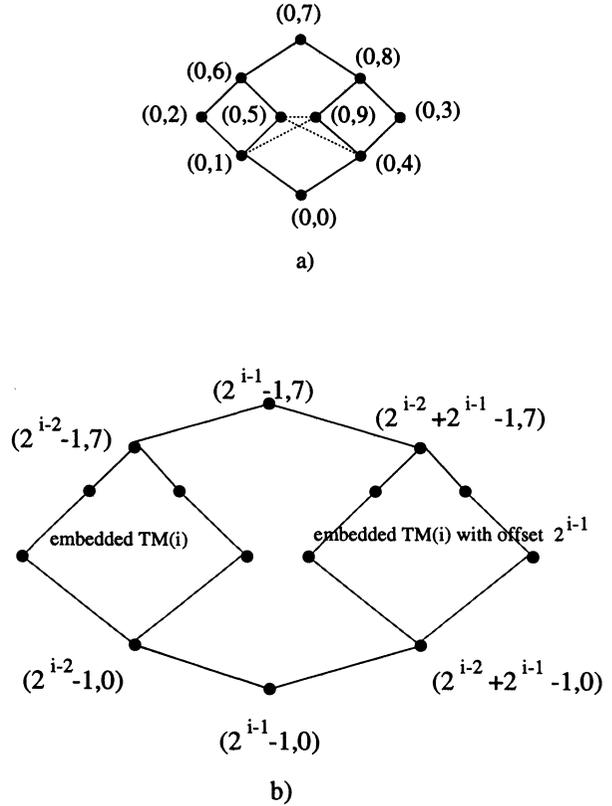


FIGURE 10 a) Embedding of $TM(2)$ in P^0 b) Modular embedding of $TM(i)$.

and erecting a copy of $CBT(n)$ along each row, using the row nodes as leaves of the tree [30].

Theorem 10: A mesh of trees $MT(n, n)$ can be embedded into HP_{2n+2} with load 1, dilation 2, edge-congestion 2 and expansion $(1.25 \cdot 2^{2n+2}) / (2^{n+1} \cdot (2^{n+1} - 2^{n-1} - 1)) \approx (5/3)$.

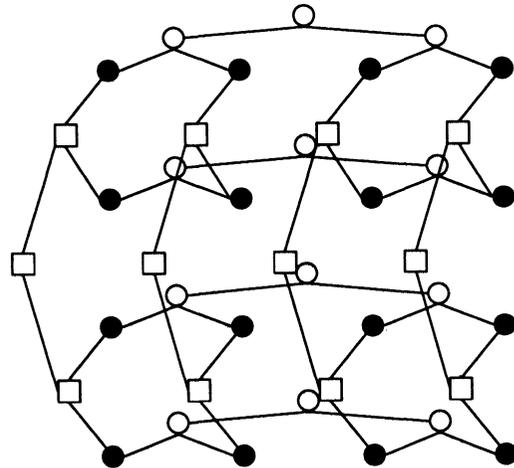


FIGURE 11 Mesh of trees $MT(2, 2)$.

Proof: Let (x, y) be a node in $MT(n, n)$, and $(\tilde{f}_n, \tilde{g}_n)$ denote the dilation-2 embedding of an X-tree $X(n)$ into HP_{n+1} as described in Section 5.5. Furthermore, let (f_n, g_n) denote the embedding of $CBT(n)$ in \mathcal{Q}_{n+1} using inorder-numbering of the nodes. Now, using the notation Π_i^z as given in Section 5.5 and writing $\text{dist}(x, y)$ as short for $\text{dist}_{HP_{2n+2}}(x, y)$, the embedding (f, g) of $MT(n, n)$ into HP_{2n+2} can be defined as follows.

$$f((x, y)) = (\Pi_1^z(\tilde{f}_n(x))f_n(y), \Pi_1^z(\tilde{f}_n(x))),$$

$$g(\{(x, y), (x', y)\}) = \begin{cases} \{f(x, y), f(x', y)\} \\ \text{if } \text{dist}(f(x, y), f(x', y)) = 1 \\ (f(x, y), (uf_n(y), v), f(x', y)) \\ \text{otherwise} \end{cases}$$

and

$$g(\{(x, y), (x, y')\}) = \begin{cases} \{f(x, y), f(x, y')\} \\ \text{if } \text{dist}(f(x, y), f(x, y')) = 1 \\ (f(x, y), (\Pi_1^z(\tilde{f}_n(x))z, \Pi_2^z(\tilde{f}_n(x))), f(x, y')) \\ \text{otherwise} \end{cases}$$

where $\tilde{g}_n(\{(x, x')\}) = (\tilde{f}_n(x), (u, v), \tilde{f}_n(x'))$, if nodes $\tilde{f}_n(x)$ and $\tilde{f}_n(x')$ are not adjacent in HP_{n+1} and $g_n(\{(y, y')\}) = (f_n(y), z, f_n(y'))$, if $f_n(y)$ and $f_n(y')$ are not adjacent in \mathcal{Q}_{n+1} .

The required values for load, dilation and edge-congestion of (f, g) follow from those of the subembeddings (f_n, g_n) and $(\tilde{f}_n, \tilde{g}_n)$. \square

Remarks:

- (1) The node-embedding f uses only $2^{n+1} \cdot (2^{n+1} - 2^{n-1} - 1)$ nodes of the HP_{2n+2} network, which has a total of $2^{2n+2} + 2^{2n}$ nodes. Since the embedding $(\tilde{f}_n, \tilde{g}_n)$ and hence (f, g) do not use the host nodes with suffix-label 1 or 4, and because the node-set of $MT(n, n)$ is only a subset of $V_{CBT(n)} \times V_{CBT(n)}$, the following $2^{2n+1} + 2^{n+1}$ nodes of HP_{2n+2} are not used by (f, g) . They are grouped into four subsets: $S_1 = \{(x(1)^{n+1}, i) \mid x \in \mathbf{Z}_2^{n-2}, i = 2, 3, 5, 9\}$, $S_2 = \{(x1, j) \mid x \in \mathbf{Z}_2^{2n-2}, j = 0, 6, 7, 8\}$, $S_3 = \{(x, j) \mid x \in \mathbf{Z}_2^{2n-1}, j \in \{1, 4\}\}$, and $S_4 = \{((1)^{n-2}x0, 0) \mid x \in \mathbf{Z}_2^n\}$.

We can use the node-set S_3 together with the node $((1)^{2n-1}, 0)$ and other nodes in S_2 for embedding two additional $CBT(2n - 1)$'s with load 1, and dilation and edge-congestion 2. The node-sets S_1 and S_4 can similarly be utilized for embedding two additional $CBT(n -$

1)'s with load 1, dilation and edge-congestion 2. The details are given in [29].

- (2) The $2^p \times 2^q$ mesh of trees $MT(p, q)$ is clearly a subgraph of $CBT(p) \times CBT(q)$. Since $HP_n = \mathcal{Q}_{n-3} \times P = \mathcal{Q}_k \times \mathcal{Q}_{n-k-3} \times P$ contains $CBT(k - 2) \times CBT(n - k - 1)$ as a subgraph for $2 < k \leq n - 1$ (and hence $CBT(1) \times CBT(n - 3)$ for $k = 2$), it follows immediately that HP_n contains a mesh of trees $MT(k - 2, n - k - 1)$ for $k > 2$ and $MT(1, n - 3)$ for $k = 2$. More precisely, we can show that two node- and edge-disjoint instances of $MT(k - 2, n - k - 1)$ can be found in HP_n , since two node- and edge-disjoint $CBT(n - 2)$'s can be embedded as subgraphs in \mathcal{Q}_n by using the subgraph embedding of a double rooted binary tree of height $n - 1$ in \mathcal{Q}_n (cf. [31]). Particularly, this implies that HP_{2n+2} contains $MT(n - 1, n)$ as a subgraph for $n > 1$ and $MT(1, 1)$ for $n = 1$. Whether $MT(n, n)$ is generally a subgraph of HP_{2n+2} for all $n \geq 1$, is still an open question.

5.9 Pyramid Embedding

The *pyramid* of height n [13], denoted by $PR(n)$, is the graph with the node-set $V_n = \{(i, x, y) \mid 0 \leq i \leq n, 0 \leq x, y \leq 2^i - 1\}$ and edge-set $E_n = \{(i, x, y), (i + 1, u, v)\} \mid 0 \leq i < n, u \in \{2x, 2x - 1\}, v \in \{2y, 2y - 1\} \cup \{(i, x, y), (i, u, v)\} \mid 0 \leq i \leq n, 0 \leq x, y \leq 2^i - 1, (x, y)$ and (u, v) differ in exactly one coordinate by one}. Informally, the pyramid network $PR(n)$ is a 4-ary complete tree of height n and the nodes at each level form a square mesh. Clearly, $P(n)$ has $N = \sum_{i=0}^n 4^i = (4^{n+1} - 1)/3$ nodes. Figure 12b) depicts a pyramid of height 2.

Theorem 11: *There exists an optimal embedding of $PR(n)$ into HP_{2n+1} with load 1, dilation 2, edge-congestion 2, and expansion 1.875.*

Proof: Since a cycle of length three is not a subgraph of HP_k for all $k \geq 3$, pyramid $PR(i)$ cannot be embedded as subgraph into HP_k for all $k \geq 2i + 1$. Theorem 11 can be proven by recursive construction, using an approach similar to that given in [13]. Figure 12a) shows that $PR(1)$ with an alternate apex can be embedded in P with load 1, edge-congestion 2, and dilation 2. For the embedding (f_i, g_i) of $PR(i)$ into HP_{2i+1} , we claim that:

- (a) f_i has load 1, dilation 2 and edge-congestion 2.
- (b) f_i maps the apex of $PR(i)$ into a node of HP_{2i+1} , that has an unassigned neighbor,

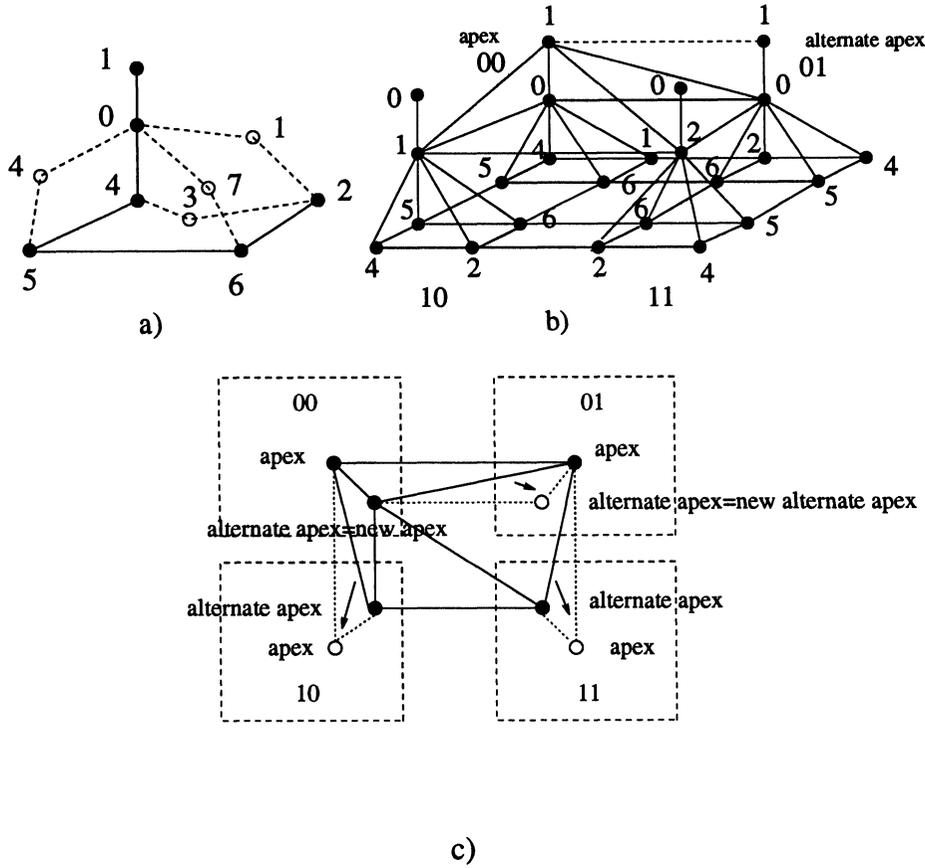


FIGURE 12 a) Embedding $PR(1)$ b) Embedding $PR(2)$ c) Embedding $PR(i + 1)$

called the *alternate apex*, such that at most one edge of $PR(i)$ is routed through the edge connecting the apex and the alternate apex.

- (c) the embedding $\tilde{f}_i(x) =$
- $$\begin{cases} f_i(\text{alternate apex}) & \text{for } x = \text{apex of } PR(i) \\ f_i(\text{apex}) & \text{for } x = \text{alternate apex of } PR(i) \\ f_i(x) & \text{for } x \neq \text{apex, alternate apex of } PR(i) \end{cases}$$
- also fulfils the properties (a) and (b).

As illustrated in Figure 12b), the nodes and edges of $PR(2)$ are embedded into HP_5 by the node-mapping f_2 and the edge routing g_2 . We embed a copy of $PR(1)$ into each of the two subgraphs P^{00} and P^{01} by the mapping f_1 and into each of P^{10} and P^{11} by \tilde{f}_1 . Then as apex of $PR(2)$ we use the alternate apex of $PR(1)$ mapped into P^{00} . Generally, the pyramid $PR(i + 1)$, for $1 \leq i \leq n - 1$, is embedded into HP_{2i+3} as follows.

Step 1: HP_{2i+3} is partitioned into four copies of HP_{2i+1}^y with the node-sets $\{(yx, i) \mid 0 \leq i \leq 9, x \in \mathbb{Z}_2^{2i-2}\}$, where $y \in \{00, 01, 10, 11\}$.

Step 2: Embed a copy of $PR(i)$ into HP_{2i+1}^{00} and HP_{2i+1}^{01} using the embedding f_i , and also embed a copy of $PR(i)$ into HP_{2i+1}^{10} and HP_{2i+1}^{11} using \tilde{f}_i as defined above.

Step 3: The apex of $PR(i + 1)$ is mapped into the alternate apex of $PR(i)$ embedded into HP_{2i+1}^{00} , and the edges are routed as sketched in Figure 12c).

The cost measurements claimed in the theorem are easy to verify. \square

Remarks: We can show the procedure of embedding $PR(n)$ into HP_{2n+1} does not use node-set $\mathcal{R} = \mathcal{R}_1 \cup \mathcal{R}_2 \cup \mathcal{R}_3$, where $\mathcal{R}_1 = \{(x, i) \mid x \in \mathbb{Z}_2^{2n-2}, i \in \{3, 7, 8, 9\}\}$, $\mathcal{R}_2 = \{(y1z, 0) \mid y \in \mathbb{Z}_2^{2n-4}, z \in \mathbb{Z}_2\}$ and $\mathcal{R}_3 = \{(x1z00,1) \mid x \in \mathbb{Z}_2^{2n-6}, z \in \mathbb{Z}_2\}$. Now, the set \mathcal{R} of nodes can be used to embed two other $PR(n)$

- 1)'s, three $PR(n - 2)$'s and one $PR(n - 3)$ in a node- and edge-disjoint fashion from the embedding of $PR(n)$, such that the load is 1, dilation and edge congestion are 2. The details are provided in [29].

sults on incomplete hypercubes are also due to us [32]. For a comprehensive outline of several embeddings on the remaining host networks, readers may refer to [33, 32, 26].

6 COMPARISON OF EMBEDDING RESULTS

This section compares the results on the embeddings of various tree-related topologies (or guest graphs) into such host graphs as hyper Petersen networks, binary hypercubes, incomplete hypercubes, de Bruijn graphs and hyper de Bruijn graphs. The results are displayed in Table 2, where the comparison metrics are dilation (\mathcal{D}), edge-congestion (\mathcal{C}) or expansion. The results listed under the hyper Petersen networks are derived in this paper. Most of the re-

7 CONCLUSIONS

In this paper, we have proposed a family of regular and highly symmetric networks, called hyper Petersen networks, and explored their suitability for use in multiprocessor interconnections. Though these networks are derived from a simple modification of the standard hypercube topology, they exhibit distinct advantages over hypercubes and some of their variants in several aspects, especially in the packing density. Besides, various topologies can be efficiently embedded into them. We have constructed

TABLE II
Comparison of Embedding Results

Guest Graph	Hyper Petersen, HP_n	Hypercube, Q_n	Incomplete Cube, I_k^n	de Bruijn, $DG(n)$	Hyper-de Bruijn, $HD(m, n)$
$CBT(n)$ Complete binary tree	$CBT(n - 1)$ in HP_n as subgraph [25] (optimal)	$CBT(n - 2)$ in Q_n as subgraph, $CBT(n - 1)$ in Q_n with $\mathcal{D} = 2$ and $\mathcal{C} = 2$ [31] (optimal)	$CBT(n - 1)$ in I_k^{n-1} as subgraph [34] (optimal)	$CBT(n - 1)$ in $DG(n)$ as subgraph [6] (optimal)	$CBT(m + n - 2)$ in $HD(m, n)$ as subgraph [21], $CBT(m + n - 1)$ in $HD(m, n)$ with $\mathcal{D} = 2, \mathcal{C} = 2$ [26]
$X(n)$ X -tree	$X(n - 1)$ in HP_n with $\mathcal{D} = 2, \mathcal{C} = 2$ (optimal) [35]	$X(n - 1)$ in Q_n with $\mathcal{D} = 2$ and $\mathcal{C} = 2$ (optimal) [35]	$X(n - 1)$ in I_k^n with $\mathcal{D} = 2, \mathcal{C} = 2$ (optimal) [32]	$X(n - 1)$ in $DG(k)$ with $\mathcal{D} = \mathcal{O}(\log n)$, expansion $\mathcal{O}(1)$ (optimal)	$X(n - 1)$ in $HD(n - 3, 3)$ with $\mathcal{D} = 2$ and $\mathcal{C} = 2$ [33]
$FRBT(n)$ Full-ringed binary tree	$FRBT(n - 1)$ in HP_n with $\mathcal{D} = 2$ and $\mathcal{C} = 2$ (optimal)	$FRBT(n - 1)$ in Q_n with $\mathcal{D} = 2$ and $\mathcal{C} = 2$ (optimal) [35]	$FRBT(n - 1)$ in I_k^n with $\mathcal{D} = 2$ and $\mathcal{C} = 2$ (optimal) [32]	$FRBT(n - 1)$ in de Bruijn with $\mathcal{C} = \mathcal{O}(1), \mathcal{D} = \mathcal{O}(\log n)$ (optimal) [26]	$FRBT(n - 1)$ in $HD(n - 3, 3)$ with $\mathcal{D} = 2, \mathcal{C} = 2$ [33]
$HT(n)$ Hypertree	$HT(n - 1)$ in HP_n with $\mathcal{D} = 2$ and $\mathcal{C} = 2$ (optimal)	$HT(n - 1)$ in Q_n with $\mathcal{D} = 2, \mathcal{C} = 2$ (optimal)	$HT(n - 1)$ in I_k^n with $\mathcal{D} = 2$ and $\mathcal{C} = 2$ (optimal) [32]	$HT(n)$ in de Bruijn host with $\mathcal{C} = \mathcal{O}(1), \mathcal{D} = \mathcal{O}(n)$ [26]	$HT(n)$ in $HD(n - 3, 3)$ with $\mathcal{D} = 2, \mathcal{C} = 2$ [33]
$TM(n)$ Tree Machine	$TM(n)$ in HP_{n+2} with $\mathcal{D} = 2, \mathcal{C} = 2$	$TM(n)$ in Q_{n+2} as subgraph [36] (optimal)	$TM(n)$ in I_k^{n+1} with $\mathcal{D} = 2, \mathcal{C} = 2$ (optimal) [32]	$TM(n)$ in $DG(n + 2)$ as subgraph [6] (optimal)	$TM(m + n - 2)$ in $HD(m, n)$ with $\mathcal{D} = 2, \mathcal{C} = 2$ [33]
$MT(n, n)$ Mesh of Trees	$MT(n, n)$ in HP_{2n+2} with $\mathcal{D} = 2, \mathcal{C} = 2$	$MT(n, n)$ in Q_{2n+2} as subgraph [36] (optimal)	$MT(n, n)$ in I_k^{2n+1} with $\mathcal{D} = 2, \mathcal{C} = 2$ [32]	$MT(n, n)$ in $DG(2, 2^{2n+2})$ with $\mathcal{D} = 2, \mathcal{C} \leq 8, \text{load} = 2$ [37]	$MT(2^p, 2^q)$ subgraph of $HD(m, n)$ for $1 \leq p \leq m - 2, 1 \leq q \leq n - 1$ [21]
$PR(n)$ Pyramid	$PR(n)$ in HP_{2n+1} with $\mathcal{D} = 2, \mathcal{C} = 2$ (optimal)	$PR(n)$ in Q_{2n+1} with $\mathcal{D} = 2, \mathcal{C} = 2$ (optimal) [3]	$PR(n)$ in I_k^{2n-1} with $\mathcal{D} = 2, \mathcal{C} = 3$ [32]	$PR(n)$ in de Bruijn host with $\mathcal{D} = \mathcal{O}(\log \sqrt{N})$ expansion $\mathcal{O}(1)$, where $N = (4^{n+1} - 1)/3$ (optimal) [26]	$PR(n)$ in $HD(2n - 2, 3)$ with $\mathcal{D} = 2, \mathcal{C} = 2$ [33]

optimal and modular embeddings of rings, meshes, hypercubes, complete binary trees, X-trees, hypertrees, tree machines, mesh of trees, and pyramids.

Although we have studied only static embeddings in the hyper Petersen networks, they are also candidates for dynamic embeddings of dynamically evolving task graphs. Therefore, another direction of our research involves dynamic embeddings of dynamically growing and shrinking data and algorithm structures such as dynamic trees and grids [38].

Work is in progress in efficiently mapping algorithms on such networks. Future research can be aimed at analyzing the performance and embedding of other topologies in the presence of faults. It would be worthwhile to design and analyze a network using multiply-twisted hypercube [9] as the framework and the Petersen graph as the building block. This will lead to further reduction in the diameter.

Recently, we have designed the folded Petersen cube networks $FPQ_{n,k} = Q_n \times FP_k$, a generalization of our approach presented in this paper, by applying iterative cartesian products on the Petersen graph [24]. It holds that $HP_n = FPQ_{n-3,1}$ is a special case of the folded Petersen cube. These networks have extremely large packing density without proportionate increase in the node-degree or diameter, yet having maximum connectivity. In fact, an folded Petersen cube network $FPQ_{n,k}$ is a regular and symmetric network consisting of $N = 2^n \cdot 10^k$ nodes with degree $\delta = 3k + n$, while the diameter is only $d = 2k + n$. Furthermore we have recently investigated incomplete folded Petersen networks having an arbitrary number of nodes [39]. Another direction of our research is to generalize the results derived for hyper Petersen or folded Petersen networks for a class of product networks in general [40].

Acknowledgments

We are grateful to the anonymous referees and the Guest Editor for their helpful comments which improved the quality of this paper.

References

- [1] J.C Bermond and C. Peyrat. de Bruijn and kautz networks: a competitor for the hypercube? In *Hypercubes and Distributed Computers*, pages 279–293, North-Holland, 1989. Elsevier Science Publishers.
- [2] F.P. Preparata and J. Vuillemin. The cube-connected cycles: A versatile network for parallel computation. *Communications of the ACM*, 24(5):300–309, May 1981.
- [3] Quentin F. Stout. Hypercubes and pyramids. In V. Cantoni and S. Levialdi, editors, *Pyramidal Systems for Computer Vision*, pages 75–89. Springer, Heidelberg, 1986.
- [4] S.K. Das. Designing recursive networks combinatorially. In *Proceedings 22nd Int. Conf. Graph Theory, Combinatorics, and Computing*, Baton Rouge, LA, February 1991.
- [5] S.B. Akers and B. Krishnamurthy. A group-theoretic model for symmetric interconnection networks. *IEEE Transactions on Computers*, 38(4), 555–566, April 1989.
- [6] M.R. Samatham and D.K. Pradhan. The de Bruijn multiprocessor network: A versatile parallel processing and sorting network for VLSI. In *IEEE Transactions on Computers*, Vol. 38, No. 4, pages 567–581, April 1989.
- [7] S. Öhring and S.K. Das. The folded Petersen network: A new communication-efficient multiprocessor topology. In *Proceedings of the 1993 International Conference on Parallel Processing, Volume 1*, pages 311–314, August 1993.
- [8] S. Abraham and K. Padmanabhan. An analysis of the twisted cube topology. In *Proc. International Conference on Parallel Processing, Vol. 1*, pp. 116–120. The Pennsylvania State University Press, 1989.
- [9] K. Efe. Programming the twisted cube architecture. In *Proc. Conf. Distri. Comput. Syst.*, pages 254–261, 1989.
- [10] A. El-Amawy and S. Latifi. Properties and performance of folded hypercubes. *IEEE Transactions on Parallel and Distributed Systems*, 2(1), 31–42, January 1991.
- [11] R.K. Das, K. Mukhopadhyaya, and B.P. Sinha. Bridged and twisted hypercubes with reduced diameters. In *1992 International Conference on Parallel Processing, Vol. 1*, pages 72–75, 1992.
- [12] G. Chartrand and L. Lesniak. *Graphs and Digraphs (Second Edition)*, Wadsworth & Brooks, 1986.
- [13] B. Monien and H. Sudborough. Embedding one interconnection network in another. In *Computational Graph Theory*, pages 257–282, Wien, 1990. Springer Verlag.
- [14] E. Horowitz and A. Zorat. The binary tree as an interconnection network: Applications to multiprocessor systems and vlsi. *IEEE Transactions on Computers*, 30, 247–253, 1981.
- [15] F.T. Leighton. Parallel computing using mesh of trees. In *Proceedings of the Workshop on Graph-Theoretic Concepts in Computer Science*, pages 200–218, 1983.
- [16] J.L. Bentley and H.T. Kung. A tree machine for searching problems. In *Proceedings International Parallel Processing Conference*, pages 257–266, 1979.
- [17] A.M. Despain and D.A. Patterson. X-tree: A tree structured multiprocessor computer architecture. In *Proceedings 5th Annual Symposium on Computer Architecture*, pages 144–151, 1978.
- [18] J.R. Goodman and C.H. Sequin. Hypertree: A multiprocessor interconnection topology. In *IEEE transactions on Computing*, volume C-30, No. 12, pages 923–933, December 1981.
- [19] H.P. Katseff. Incomplete hypercubes. *IEEE Transactions on Computers*, 37(5), 604–608, May 1988.
- [20] N.G. de Bruijn. A combinatorial problem. In *Proceedings*, volume 49, part 20, pages 758–764. Akademie Van Wetenschappen, 1946.
- [21] E. Ganesan and D.K. Pradhan. The hyper-de Bruijn multiprocessor networks. In *Proceedings of the 11th Conference on Distributed Computing Systems*, pages 492–499, Arlington, TX, May 1991.
- [22] A.H. Esfahanian, L.M. Ni, and B.E. Sagan. On enhancing hypercube multiprocessors. In *Proceedings of the International Conference on Parallel Processing*, pages 86–89, 1988.
- [23] G. Chartrand and R.J. Wilson. The Petersen graph. *Graphs and Applications* (Eds. F. Harary and J.S. Maybee), pages 69–100, 1985.
- [24] S. Öhring and S.K. Das. The folded Petersen cube network: New competitors for the hypercubes. Technical Report CRPDC-93-8, University of Wuerzburg, University of North Texas, June 1993.
- [25] S.K. Das and A.K. Banerjee. Hyper Petersen network: Yet another hypercube-like topology. In *Proceedings of the 4th Symposium on the Frontiers of Massively Parallel Compu-*

- tation (*Frontiers '92*), pages 270–277, McLean, Virginia, USA, October 1992. Computer Society Press.
- [26] S. Öhring. Optimal dynamic embeddings of arbitrary trees in de Bruijn graphs. In *Proceedings of the Fourth Italian Conference Theoretical Computer Science*, pages 279–293. World Scientific, October 1992.
- [27] S.N. Bhatt, F.R.K. Chung, J.-W. Hong, F.T. Leighton, B. Obrenić, A.L. Rosenberg, and E.J. Schwabe. Optimal emulations by butterfly-like networks. COINS Technical Report 90-108, Computer and Information Science Department, University of Massachusetts, 1990.
- [28] A.S. Wagner. Embedding all binary trees in the hypercube. In *Proceedings of the Third IEEE Symposium Parallel and Distributed Processing*, pages 104–111, Dallas, TX, December 1991.
- [29] S.K. Das and S. Öhring. Embeddings of tree-related topologies in hyper Petersen networks. Technical Report CRPDC-92-16, University of North Texas, University of Wuerzburg, September 1992.
- [30] A.L. Rosenberg. Product-shuffle networks: Towards reconciling shuffles and butterflies. COINS Technical Report 89-113, Computer and Information Science Department, University of Massachusetts, 1989.
- [31] S.N. Bhatt and I. Ipsen. How to embed trees in hypercubes, tech. report rr-443. Technical report, Department of Computer Science, Yale University, New Haven, CT. 1985.
- [32] S. Öhring and S.K. Das. Embeddings of tree-related networks in incomplete hypercubes. In *Proceedings of Parallel Architectures and Languages Europe 1993, Munich, Germany in Lecture Notes in Computer Science, Vol. 694*, pages 80–91, June 1993.
- [33] S. Öhring and S.K. Das. Dynamic embeddings of trees and quasi-grids into hyper-de Bruijn networks. In *Proceedings of the 7th International Parallel Processing Symposium, Newport Beach, California*, pages 519–523, April 1993.
- [34] N.F. Tzeng, H.-L. Chen, and P.-J. Chuang. Embeddings in incomplete hypercubes. In *Proceedings of the 1990 International Conference on Parallel Processing, Vol. III*, pages 335–339, 1990.
- [35] C.-T. Ho. *Optimal Communication Primitives and Graph Embeddings on Hypercubes*. PhD thesis, Yale university, May 1990.
- [36] K. Efe. Embedding mesh of trees in the hypercube. *Journal of Parallel and Distributed Computing*. 11:222–230, 1991.
- [37] E.J. Schwabe. Embedding meshes of trees into deBruijn graphs. *Information Processing Letters*, 43, 237–240, October 1992.
- [38] S. Öhring and S.K. Das. Mapping dynamic data and algorithm structures into product networks. In *Proceedings of the 4th International Symposium on Algorithms and Computation (ISAAC'93), Hong Kong*, December 15–17, 1993, Lecture Notes in Computer Science, vol. 762, Springer Verlag, pp. 147–156.
- [39] S. Öhring, D.H. Hohndel and S.K. Das. Scalable Interconnection Networks based on the Petersen graph. In *Proceedings of the 7th International Conference on Parallel and Distributed Computing Systems (PDCS '94), Las Vegas, Nevada*, pp. 581–586, October, 1994.
- [40] S. Öhring and D.H. Hohndel. Optimal Fault Tolerant Communication Algorithms on Product Networks Using Spanning Trees, to be published in *Proceedings of the 6th IEEE Symposium on Parallel and Distributed Processing*, October 1994.

Biographies

SAJAL K. DAS received the Ph.D. degree in Computer Science in 1988 from the University of Central Florida, Orlando. Currently he is an Associate Professor of Computer Science at the University of North Texas, Denton, and also a faculty of the Center for Research in Parallel and Distributed Computing. A recipient of *Honor Professor Award* at UNT for best teaching and scholarly research, he was invited in 1993 to visit *The Leonardo Fibonacci Institute* in Trento, Italy. His research interests include design and analysis of parallel algorithms, parallel data structures, parallel discrete event simulation, applied graph theory and combinatorics, and multiprocessor interconnection networks. He has published over 50 journal papers in these areas and presented his research work at many international and national conferences. He serves on the Editorial Boards of *Parallel Processing Letters* (World Scientific Pub) and *Journal of Parallel Algorithms and Applications* (Gordon and Breach). He has been a member of Program Committees for several international conferences. He is also a member of IEEE Computer Society, Association of Computing Machinery, New York Academy of Sciences, and Sigma Xi.

SABINE ÖHRING received the Diploma degree in mathematics and computer science in 1991 and the PhD degree in Computer Science in 1994 from the University of Würzburg in Germany. Currently, she is a Research Associate at the Department of Computer Science of the University of North Texas, Denton. Her research interests include graph embeddings, the design and analysis of interconnection networks for multicomputer systems, communication on interconnection networks, parallel data structures and parallel discrete event simulation. She is a member of IEEE Computer Society, Association of Computing Machinery, EATCS, 61 and PARS.

AMIT K. BANERJEE received the B. Tech. degree in Radiophysics and Electronics from Calcutta University in 1985 and the M.S. degree in Computer Science in 1992 from the University of North Texas, Denton. Currently he is working for the National Airport Authority, Guwahati, in India. His research interest is in interconnection networks and multiprocessor architectures.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

