Construction of Large-Size Interconnection Networks with High Performance*

Hong Shen† and Ralph-Johan Back

Department of Computer Science, Åbo Akademi University, Lemminkäisenkatu 14, SF-20520 Turku, Finland

This paper proposes a new method, *recursive expansion* (RE), for systematically constructing interconnection networks of arbitrary large size with high performance. On the basis of two small-size networks, a frame and a unit, the RE method works in a manner of recursively replacing each node in the frame with an expanded network containing a set of copies of the unit and each edge in the frame with a set of interunit connections connecting a pair of the networks until a network of the desired size has been obtained. By RE, we can construct various kinds of large-size and low-cost interconnection networks. Two applications of the method, the $\mathcal{T}_r^{\Sigma}$ network based on the torus and the $\mathcal{H}_r^{\Sigma}$ network based on the hypercube, show that our method can produce networks with cost $O((\log^{3/2} n)/(\log^{3/2} \log n))$ (degree O(1)) and $O(\log n \log \log n)$ (degree O(log log n)). In addition to low cost, networks constructed by RE also possess other properties such as high constructability, good extendability, symmetric topology, and efficient message routing. This paper describes an algorithm, for automatically constructing arbitrary large size networks with high performance. For constructing a network of size n_r through r phases RE on the basis of a frame of degree d_f and a unit of size n_u and degree d_u, the algorithm has a time complexity $O((\max\{(d_f/n_u), (d_u/r)\})rn_r^2)$. Finally, a routing algorithm for networks constructed by RE is presented. The routing algorithm can realize point-to-point message routing without using a global routing table at each node and has a time complexity $O((k_u + k_f)d_f r)$, where k_u and k_f are diameters of the frame and of the unit, d_f is the degree of the frame, and r is the number of phases of RE to construct the network.

1. INTRODUCTION

The appearance of multiprocessor systems whose processors communicate to each other through an *interconnection network* is one of the main contemporary developments in computer science and technology. The interconnection network is a key part in a multiprocessor system and its configuration decides the hardware structure of the system [2]. How to construct high-performance interconnection networks is a fundamental task in multiprocessor-system design. For designing massive parallel systems containing a large number of processors (more than 1000), this problem becomes very important.

An interconnection network can be represented by a graph $G(V, E)$ consisting of a set of nodes (V) and a set of edges (E) connecting the nodes. Without loss of generality, we assume the graph to be undirected. The size of a network is the number of the nodes of the network. A network is said *large* (*small*) if its size is large (small). The *node-degree* of a node in a network is the number of edges incident to the node. The *distance* between two nodes in a network is the length of

*This work was partially supported by the FINSOFT III Research Program

†Current affiliation: School of Computing and Information Technology, Griffith University, Nathan, QLD 4111, Australia.

NETWORKS, Vol. 23 (1993) 399–414

CCC 0028-3045/93/404399-16

the shortest path between them. There are different measures for the performance of an interconnection network. Two key parameters are the *network-degree,* i.e., the maximum node-degree over all nodes in the network, and the *network-diameter,* i.e., the maximum distance over all pairs of nodes in the network. Other aspects are the network *extendability* (*scalability*), *simplicity* of message-routing, traffic *uniformity,* network *reliability,* and so on [1, 6]. A typical performance measure for a *static* interconnection network [5] is the product of the network-degree and the network-diameter [8], which we call the *cost* of the network. The degree indicates the hardware costs for constructing the network and diameter shows the maximum delay for message routing in the network. A *high-performance* network should keep its cost as small as possible. For a fixed-size network, the degree and diameter of the network are often contradictory: Decreasing one will cause increasing the other. There is usually a trade-off between them. Accordingly, for a fixed value degree and diameter, The network size is bounded by Moore's theorem [4].

Designing high-performance interconnection networks has been extensively studied in the literature [2, 5, 7]. There are many elegant networks that have been widely used in variant multiprocessor systems such as *hypercube, mesh,* and *cube-connected-in-cycles* [1, 7]. However, those previous results are concerned mainly with designing a network of a specific topology rather than with general design methods.

This paper will focus on a general method for designing high-performance interconnection networks of arbitrary large size. The main contributions of the paper are the following:

1. We propose a systematic method for interconnection network design, *recursive expansion* (RE), by which various kinds of large networks with a high performance can be constructed on the basis of two given small networks, a frame, and a unit.
2. We exhibit the performance of the networks constructed by recursive expansion.
3. We show two applications of recursive expansion: the $\mathcal{T}_r^{\Sigma}$ network of degree O(1) and diameter $O((\log^{3/2} n)/(\log^{3/2} \log n))$, and the $\mathcal{H}_r^{\Sigma}$ network of degree O(log log n) and diameter O(log n log log n), where n is the size of the networks.
4. We present an efficient algorithm for automatic construction of large networks by recursive expansion.
5. We present an efficient message routing algorithm for networks constructed by recursive expansion.

An earlier version of this paper was presented at EUROMICRO 92 Conference.

We use the term *network* instead of *interconnection network* in the following sections.

2. THE RECURSIVE EXPANSION METHOD

A small network is obviously much easier to construct to reach high performance than is a large network. Recursively expanding the network size without losing the network performance on the basis of some chosen small networks with desired performance is a promising way to obtain a large network with high performance. This motivates our method for constructing large networks, *recursive expansion* (RE for short).

Before describing the RE method, we need to introduce the following definitions and notations:

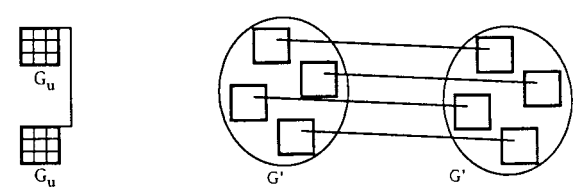
Definition 1. Two networks $G(V, E)$ and $G'(V', E')$ are *disjoint* if $V \cap V' = \emptyset$ and $E \cap E' = \emptyset$. All of m ($m > 2$) networks are *disjoint* if any two of them are disjoint. The *disjoint union* [4] of m copies of a network $G(V, E)$, denoted by m G-copies or mG, is a network consisting of m disjoint networks, each a copy of (i.e., isomorphic to) G.

Definition 2. A *unit,* denoted by G_u, is a (small) network. We say that an edge connects two networks if it connects a node in one network to a node in the other network. Let $2G_u \subset G$. Any edge connecting the two copies of G_u is referred to as an *interunit connection* for the $2G_u$. Similarly, assuming $2G' \subset G$ and $mG_u \subset G'$, we call a set of interunit connections, one-to-one connecting the unit-copies in one G' to the corresponding unit-copies in the other G', and *interunit connection set* for all unit copies between the $2G'$ in G.

Figure 1 shows an example of an interunit connection (a) and an interunit connection set (b).

Definition 3. Two networks are said to be *adjacent* if two adjacent nodes in another network are replaced by these two networks so that the edge incident to the two nodes is replaced with a set of edges connecting each pair of two corresponding nodes of the two networks.

The basic idea of the recursive expansion method is the following:



(a) An inter-unit connection (b) An inter-unit connection set

Fig. 1. An example of connections on unit copies.

- Take two small networks, G_f and G_u, with the desired performance and properties, where G_f is a *frame* that describes the topological structure of the expanding space and G_u is a unit that gives the topology of all unit copies during the expanding.
- Copy G_f into G_1, replace each node in G_1 with a G_u copy (all copies are disjoint) and each edge in G^1 with an interunit connection between two adjacent G_u copies; thus get the first-phase expanded network G_1 consisting of a set of G_u copies together with a set of interunit connections.
- Again copy G_f into G_2, replace each node in G_2 with a G_1 copy (all copies are disjoint) and each edge with an interunit connection set between two adjacent G_1 copies; thus get the second-phase expanded network G_2 consisting of a set of G_1 copies together with a set of interunit connections.
- Repeat the above expanding until a network of the required size is obtained.

We use disjoint node sets, namely, *pivot sets*, in each unit copy to realize the interunit connections in different expansion phases. The interunit connections in one phase of expansion are all incident to the nodes in one pivot set of each unit copy. For instance, we may choose the pivot sets in such a way that each set only contains a single node. In Figure 2, we show how to construct a large network, G_3, of 192 nodes and degree 4 on the basis of a fram G_f of 4 nodes and degree 2 and a unit G_u of 3 nodes and degree 2 by applying the RE method, where p_i indicates the pivot node in each unit copy to which the interunit connections are incident in the i-th phase of expansion, $1 \le i \le 3$.

Now, we give the formal description of the RE method. Assume that we need r phases expanding on the basis of frame G_f and unit G_u to obtain a network, G_r, of the required size by the above RE.

Notation 1. We write (n, d, k)-G for a network, G, of size n, degree d, and diameter k.

Notation 2. We write $\{(n_f, d_f, k_f)\text{-}G_f, (n_u, d_u, k_u)\text{-}G_u\} \xrightarrow{\text{RE}_r} (n_r, d_r, k_r)\text{-}G_r$ for the r phases RE from (n_f, d_f, k_f)-G_f and (n_u, d_u, k_u)-G_u to (n_r, d_r, k_r)-G_r, where the ith-phase RE produces G_i, $1 \le i \le r$.

We also use the notation "$|P|$" for the size of set P, i.e., the number of elements in P.

Definition 4. For $\{(n_f, d_f, k_f)\text{-}G_f, (n_u, d_u, k_u)\text{-}G_u\} \xrightarrow{\text{RE}_r} (n_r, d_r, k_r)\text{-}G_r$ the *pivot-set sequence*$_{(s)}$, PSS$_{(s)}$ for short, is a sequence of s disjoint node sets in G_u, P_1, P_2, $\cdots$, P_s, which meets the following conditions:

1. For any two nodes of P_i, there is a path in G_u on which all nodes are in P_i, $1 \le i \le s$.

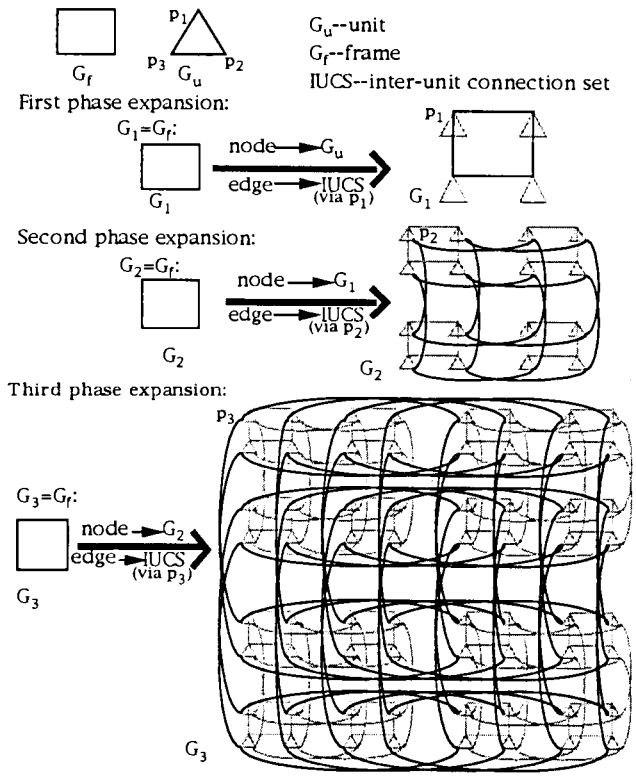


Fig. 2. Construction of large networks by recursive expansion (RE).

2. $|P_i| = |P_j|$, $1 \le i, j \le s$.
3. $1 \le |P_i| \le \min\{d_f, n_u/s\}$, $1 \le i \le s$.

New edges used for the interunit connection during the ith-phase RE are assigned to nodes in $P_{1+(i-1) \bmod(n_u)}$ as evenly as possible, $1 \le i \le r$.

We call the subnetwork in G_u the region of pivot set P_i, denoted by $R(P_i)$, $1 \le i \le s$, if its nodes are all nodes in P_i and edges are all edges incident to those nodes.

We write "PSS$_{(s_d)}$" and "PSS$_{(s_k)}$" for PSS$_{(s)}$ used, respectively, for minimizing the degree and diameter of G_r.

Clearly, for PSS$_{(s_d)}$, minimizing the degree of G_r can be realized by making the size of each pivot set as close as possible to d_f so that the number of new edges for interunit connection assigned to each node in the pivot set is as small as possible (the assignment of new edges to nodes in each pivot set is in a way as even as possible, by Definition 3), whereas for PSS$_{(s_k)}$, minimizing the diameter of G_r can be realized by setting the size of each pivot set to 1, the minimum size, so that the new edges in each phase of RE will be all assigned to the only node in each pivot set, which makes the inter-unit communication to be directly realized by the internode communication among all nodes of the pivot sets to this phase and therefore eliminates any mes-

sage passing between any nodes within each pivot set. Hence, we have the following theorem:

Theorem 1. For $\{(n_f, d_f, k_f)\text{-}G_f, (n_u, d_u, k_u)\text{-}G_u\} \xrightarrow{\text{RE}^r} (n_r, d_r, k_r)\text{-}G_r$, where $\lambda n_u < r \le (\lambda + 1)n_u$ and λ is an integer, $\text{PSS}_{(s_d)}$ and $\text{PSS}_{(s_k)}$ meet the following equations:

1. For $\text{PSS}_{(s_d)}$,
   (a)
   $$s_d = \begin{cases} n_u & \text{for RE at phase } i,\ 1 \le i \le \lambda n_u \\ r - \lambda n_u & \text{for RE at phase } i,\ \lambda n_u < i \le r; \end{cases}$$
   (b) $|P_i| = \min\{\lfloor n_u/s_d \rfloor, d_f\},\ 1 \le i \le s_d$.
2. For $\text{PSS}_{(s_k)}$,
   (a)
   $$s_k = \begin{cases} n_u & \text{if } \lambda > 0 \\ r & \text{if } \lambda = 0; \end{cases}$$
   (b) $|P_i| = 1,\ 1 \le i \le s_k$.

Proof. The correctness of the theorem is trivial by Definition 4 and the meaning of $\text{PSS}_{(s_d)}$ and $\text{PSS}_{(s_k)}$. ■

By Theorem 1, we know that $PSS_{(s_d)}$ has two appearances during r phases RE ($\lambda n_u < r \le (\lambda + 1)n_u$), one for the first λn_u phases that is the same as $\text{PSS}_{(s_k)}$ and the other for the final $r - \lambda n_u$ phases. The two appearances can be the same if $r - \lambda n_u > \lfloor n_u/2 \rfloor$. This shows that the construction of G_r by taking $\text{PSS}_{(s_d)}$ will be more complicated than that by taking $\text{PSS}_{(s_k)}$. Also, the interunit communication through nodes of the pivot sets in $\text{PSS}_{(s_d)}$ is more complicated than that in $\text{PSS}_{(s_k)}$ because of the need of internode message passing within the region of a pivot set. However, because taking $\text{PSS}_{(s_d)}$ can achieve a smaller degree of target network G_r than taking the $\text{PSS}_{(s_k)}$, it has a great value from the point of view of hardware expense. The real choice of $\text{PSS}_{(s_d)}$ or $\text{PSS}_{(s_k)}$ shall, however, depend on the requirements in the real case.

For $\{(n_f, d_f, k_f)\text{-}G_f, (n_u, d_u, k_u)\text{-}G_u\} \xrightarrow{\text{RE}^r} (n_r, d_r, k_r)\text{-}G_r$, we form the pivot-set sequence in G_u, $P_1, P_2, \cdots, P_s$, by Definition 4. Thus, the interunit connections between each pair of adjacent G_{i-1} copies in the ith-phase RE are realized by one-to-one connecting nodes in different $P_{1+(i-1)\bmod(n_u)}$ of all unit copies in one G_{i-1} copy to the corresponding ones in the other G_{i-1} copy, where $1 \le i \le r$ and $G_0 = G_u$. This is illustrated by Figure 3 as follows:

Our RE method can be generally described as the following algorithm:

Algorithm RE(n_r, $\mathcal{R}$)

{*Construct a network of size n_r with property requirements $\mathcal{R}$.*}

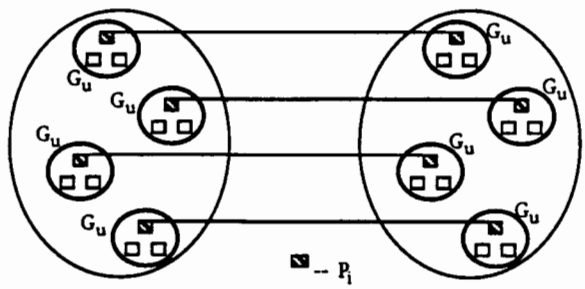


Fig. 3. Interunit connection in the ith-phase RE via pivot-node set P_i.

1. Choose a $(n_f, d_f, k_f)\text{-}G_f$ and $(n_u, d_u, k_u)\text{-}G_u$ as the frame and unit, respectively, according to $\mathcal{R}$, where $n_u n_f^{r-1} < n_r \le n_u n_f^r$.
2. Form the pivot-set sequence, $P_1, P_2, \cdots, P_s$, in G_u.
3. For $i := 1$ to r do
   Copy G_f into G_i;
   One-to-one replace all nodes in G_i with disjoint, G_{i-1} copies, all edges in G_i with interunit connection sets (each for a pair of G_{i-1} copies, realized by one-to-one connecting nodes in all different $P_{1+(i-1)\bmod(n_u)}$ in one G_{i-1} to nodes in the corresponding $P_{1+(i-1)\bmod(n_u)}$ in the other G_{i-1}.

3. PERFORMANCE ANALYSIS OF THE TARGET NETWORK

For $\{(n_f, d_f, k_f)\text{-}G_f, (n_u, d_u, k_u)\text{-}G_u\} \xrightarrow{\text{RE}^r} (n_r, d_r, k_r)\text{-}G_r$, the relation of network size, degree, and diameter between the target network G_r and the initial networks G_f and G_u follows the following theorem:

Lemma 1. For $\{(n_f, d_f, k_f)\text{-}G_r, (n_u, d_u, k_u)\text{-}G_u\} \xrightarrow{\text{RE}^r} (n_r, d_r, k_r)\text{-}G_r$, the size of G_r meets the following equation:

$$n_r = n_u n_f^r. \tag{1}$$

Proof. The correctness of the lemma is trivial by the fact that the size of the result network after each phase of RE will be as large as n_f times of that before the phase. ■

Lemma 2. For $\{(n_f, d_f, k_f)\text{-}G_r, (n_u, d_u, k_u)\text{-}G_u\} \xrightarrow{\text{RE}^r} (n_r, d_r, k_r)\text{-}G_r$, where $\lambda n_u < r \le (\lambda + 1)n_u$ and λ is an integer, the degree of G_r meets the following formulae:

1. If $\text{PSS}_{(s_k)}$ is taken,
$$d_r \le d_u + (\lambda + 1)d_f. \tag{2}$$

2. If $\mathrm{PSS}_{(s_d)}$ is taken,

$$d_r \le d_u + \lambda d_f + \left\lceil \frac{d_f}{\left\lfloor \frac{n_u}{r - \lambda n_u} \right\rfloor} \right\rceil. \tag{3}$$

Proof. Obviously, d_r will never be greater than the sum of the maximum node degree of all nodes in G_u, d_u, and the maximum number of new edges possibly added to a node in any pivot set in G_u during r phases RE that is assumed to be d_Δ, i.e.:

$$d_r \le d_u + d_\Delta.$$

Thus, Eq. (2) is trivial by Definition 4 and Theorem 1.

When $\mathrm{PSS}_{(s)}$ in G_r is $\mathrm{PSS}_{(s_d)}$, by Theorem 1 we know that there are s_d pivot sets in the pivot-set sequence, $P_1, P_2, \cdots, P_{(s_d)}$, where $s_d = n_u$ for the first λn_u phases RE and $s_d = r - \lambda n_u$ for the final $r - \lambda n_u$ phases RE. Because $|P_i| = |P_j|$ $(i \ne j, 1 \le i, j \le s_d)$, we have

$$|P_i| = \begin{cases} 1 & \\ & 1 \le i \le n_u, \text{ for the first } \lambda n_u \text{ phases RE} \\ \min\left\{\left\lfloor \frac{n_u}{r - \lambda n_u} \right\rfloor, d_f\right\} & \\ & 1 \le i \le r, \text{ for the final } r - \lambda n_u \text{ phases RE.} \end{cases}$$

Because the size of each pivot set is not greater than d_f, the degree of G_f, and the new edges for interunit connection assigned to nodes in $P_{1+i \bmod(n_u)}$ in the ith-phase RE are distributed as evenly as possible, $1 \le i \le r$, by Definition 4, the maximum number of new edges assigned to each node in any unit copy in the expanded network during the ith-phase RE, $d_\Delta^{(i)}$ is

$$d_\Delta^{(i)} = \left\lceil \frac{d_f}{|P_i|} \right\rceil \le \begin{cases} d_f & \text{for } 1 \le i \le \lambda n_u \\ \left\lceil \frac{d_f}{\left\lfloor \frac{n_u}{r - \lambda n_u} \right\rfloor} \right\rceil & \text{for } \lambda n_u < i \le (\lambda + 1) n_u. \end{cases}$$

Furthermore, because the new edges for the interunit connection in the ith-phase RE are assigned to nodes in $P_{1+(i-1) \bmod(n_u)}$, $1 \le i \le r$, also for any α and β, where $\alpha \ne \beta$ and $1 \le \alpha, \beta \le s_d$, $P_\alpha \cap P_\beta = \emptyset$, we have that during the first λn_u phases RE there are λ phases in which the new edges for interunit connection will be assigned to nodes in each pivot set and that during the final $r - \lambda n_u$ phases RE nodes in each pivot set only accept new edges for one phase. Therefore,

$$d_\Delta = \begin{cases} \lambda d_f & \text{for the first } r - \lambda n_u \text{ phases RE} \\ \left\lceil \frac{d_f}{\left\lfloor \frac{n_u}{r - \lambda n_u} \right\rfloor} \right\rceil & \text{for the final } r - \lambda n_u \text{ phases RE.} \end{cases}$$

This is equivalent to the following equation:

$$d_\Delta = \lambda d_f + \left\lceil \frac{d_f}{\left\lfloor \frac{n_u}{r - \lambda n_u} \right\rfloor} \right\rceil, \quad \text{for the whole } r \text{ phases.}$$

Hereby, the lemma holds. ■

Lemma 3. For $\{(n_f, d_f, k_f)\text{-}G_r, (n_u, d_u, k_u)\text{-}G_u\} \xrightarrow{\mathrm{RE}'} (n_r, d_r, k_r)\text{-}G_r$, where $\lambda n_u < r \le (\lambda + 1) n_u$ and λ is an integer, the diameter of G_r meets the following formulae:

1. If $\mathrm{PSS}_{(s_k)}$ is taken,

$$k_r = k_u(r + 1) + k_f r. \tag{4}$$

2. If $\mathrm{PSS}_{(s_d)}$ is taken,

$$k_r = k_u(r + 1) + k_f(d_f r - (d_f - 1)\lambda n_u) - (d_f - 1)(r - \lambda n_u). \tag{5}$$

Proof. Let G_i be the expanded network after i phases RE on the basis of G_f and G_u, $1 \le i \le r$. Clearly, by RE, G_r consists of a set of G_{r-1} copies: $G_{r-1}^{(1)}, G_{r-1}^{(2)}, \cdots, G_{r-1}^{(n_f)}$, where $G_{r-1}^{(j)}$, is the G_{r-1} copy replacing node j of G_f, $1 \le j \le n_f$, together with an interunit connection set between adjacent G_{r-1} copies, assuming $G_0 = G_u$. Let x and y be two nodes in G_r. Without loss of generality, assume that x belongs to unit copy $G_u^{(s,\tilde{x})}$ in $G_{r-1}^{(s)}$ and y belongs to $G_{r-1}^{(t)}$, where $1 \le \tilde{x} \le n_f^{r-1}$, $1 \le s, t \le n_f$, and $s \ne t$. $G_u^{(t,\tilde{x})}$ is the corresponding unit copy in $G_{r-1}^{(t)}$ to $G_u^{(s,\tilde{x})}$ in $G_{r-1}^{(s)}$. We also write $P_r^{(s,\tilde{x})}$ and $P_r^{(t,\tilde{x})}$ for pivot set P_r in $G_u^{(s,\tilde{x})}$ of $G_{r-1}^{(s)}$ and in $G_u^{(t,\tilde{x})}$ of $G_{r-1}^{(t)}$, respectively, $1 \le \tilde{x} \le n_f^{r-1}$, $1 \le i, j \le n_f$.

Let the distance (shortest path) between two networks be the minimum distance (shortest path) between all nodes of one network and all nodes of the other network. From RE, we know that if two G_{r-1} copies are adjacent, all unit copies of them are pairwise adjacent via an interunit connection set connecting nodes in P_r of all pairs of corresponding unit copies. Therefore, between any pair of G_{r-1} copies in G_r there are a number of shortest paths pairwise connecting (nodes in P_r of) all corresponding unit copies of them. Let p be such a node in $P_r^{(s,\tilde{x})}$ that there is a shortest path between $G_{r-1}^{(s)}$ and $G_{r-1}^{(t)}$ that connects p to

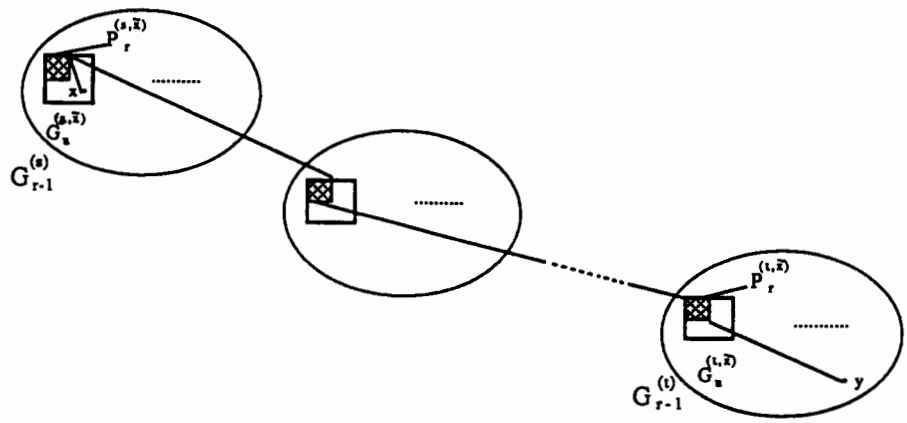


Fig. 4. The distance between two nodes x and y in G_r.

a node q (in $P_r^{(t,\bar{x})}$) in $G_{r-1}^{(t)}$. The distance between x and y, denoted by $dist(x, y)$, can be divided into three consequential parts as shown in Figure 4: the distance between x and y, denoted by $dist(x, p|p \in P_r^{(s,\bar{x})})$, the distance between p and q, denoted by $dist(p|p \in P_r^{(s,\bar{x})}, q|q \in P_r^{(t,\bar{x})})$, and the distance between q and y, denoted by $dist(q|q \in P_r^{(t,\bar{x})}, y)$; i.e.:

$$\begin{aligned} dist(x, y) = {} & dist(x, p|p \in P_r^{(t,\bar{x})}) \\ & + dist(p|p \in P_r^{(s,\bar{x})}, q|q \in P_r^{(t,\bar{x})}) \\ & + dist(q|q \in P_r^{(t,\bar{x})} y). \end{aligned} \quad (6)$$

We choose x and y such that the distance between x and y is the maximum distance of all pairs of nodes in G_r, the diameter of G_r, as follows:

$$\begin{aligned} k_r = {} & \max_{x,y} \{dist(x, y)\} = \max_x \{dist(x, p|p \in P_r^{(s,\bar{x})})\} \\ & + \max_{G_{r-1}^{(t)}} \{dist(p|p \in P_r^{(t,\bar{x})}, q|q \in P_r^{(t,\bar{x})})\} \\ & + \max_y \{dist(q|q \in P_r^{(t,\bar{x})}, y)\}. \end{aligned} \quad (7)$$

Because x and p belong to the same unit copy $G_u^{(s,\bar{x})}$ in $G_{r-1}^{(s)}$, the first part of the above equation meets

$$\max_x \{dist(x, p|p \in P_r^{(s,\bar{x})})\} = k_u. \quad (8)$$

By Definition 2 and Algorithm RE, we know that each pair of adjacent G_{r-1} copies in G_r with respect to the topology of G_f are connected by one-to-one connecting (nodes in P_r of) all unit copies in one G_{r-1} copy to the corresponding ones in the other G_{r-1} copy. All n_f G_{r-1} copies in G_r are connected in the topology of G_f with one G_{r-1} copy and one interunit connection set corresponding, respectively, to one node and one edge in G_f. First, because the diameter of G_f is k_f, taking each P_r as a single node and without considering any message passing intra-$R(P_r)$ will easily get that any node in P_r of each unit copy in G_{r-1}^{α} can reach one node in one P_r of the corresponding unit copy in G_{r-1}^{β}, where $\alpha \neq \beta$ and $1 \leq \alpha, \beta \leq n_f$, with a maximum distance k_f. Consequently, considering the real case that P_r may contain more than one node and that message passing is therefore carried out first intra-$R(P_r)$ and then inter-$R(P_r)$, we get that any node in each P_r of G_{r-1}^{α} can reach one node in the corresponding P_r of G_{r-1}^{β} with a maximum distance $(|P_r| - 1)k_f + k_f$, where $\alpha \neq \beta$ and $1 \leq \alpha, \beta \leq n_f$. Because p is the correct node initiating a shortest path from $G_{r-1}^{(s)}$ to $G_{r-1}^{(t)}$, there is no intra-$R(P_r)$ message passing in ($R(P_r^{(s,\bar{x})})$ of) $G_{r-1}^{(s)}$. Thus, we get the second part in Eq. (7) as

$$\begin{aligned} \max_{G_{r-1}^{(t)}} \{dist(p|p \in P_r^{(s,\bar{x})}, q|q \in P_r^{(t,\bar{x})})\} = {} & (|P_r| - 1)(k_f - 1) \\ & + k_f = |P_r|(k_f - 1) + 1. \end{aligned} \quad (9)$$

The third part of Eq. (7), i.e., the maximum distance between the known node q and an arbitrary node y in $G_{r-1}^{(t)}$, is clearly equal to the diameter of $G_{r-1}^{(t)}$: k_{r-1}.

Thus, we get

$$\begin{aligned} & k_0 = k_u, \\ & k_r = k_u + |P_r|(k_f - 1) + 1 + k_{r-1}, \quad r \geq 1. \end{aligned} \quad (10)$$

The solution of Eq. (10) is

$$k_r = k_u(r + 1) + ((k_f - 1)|P_r| + 1)r. \quad (11)$$

By Definition 4 and Theorem 1, we have that if $PSS_{(s)}$ is $PSS_{(s_k)}$, then $|P_r| = 1$, and if it is $PSS_{(s_d)}$, then $|P_r| = 1$ for the first λn_u RE and $|P_r| \leq d_f$ for the final $r - \lambda n_u$ RE. Thus, Eqs. (4) and (5) hold. This completes the proof. ■

Hence, we have the following theorem:

Theorem 2. For $\{(n_f, d_f, k_f)\text{-}G_f, (n_u, d_u, k_u)\text{-}G_u\} \xrightarrow{\text{RE}^r} (n_r, d_r, k_r)\text{-}G_r$, where $\lambda n_u < r \le (\lambda + 1)n_u$ and λ is an integer, the size, degree, and diameter of G_r meet the following formulae:

1. $n_r = n_u n_f^r$.
2. If $\text{PSS}_{(s_k)}$ is taken,
   (a) $d_r \le d_u + (\lambda + 1)d_f$;
   (b) $k_r = k_u(r + 1) + k_f r$.
3. If $\text{PSS}_{(s_d)}$ is taken,
   (a) $d_r \le d_u + \lambda d_f + \left\lceil \dfrac{d_f}{\left\lfloor \dfrac{n_u}{r - \lambda n_u} \right\rfloor} \right\rceil$;
   (b) $k_r = k_u(r + 1) + k_f(d_f r - (d_f - 1)\lambda n_u) - (d_f - 1)(r - \lambda n_u)$.

Theorem 2 shows the relations of network size, degree, and diameter between the target network G_r and the initial networks G_f and G_u for RE in the case of unbounded (number of phases of) expanding. Now, we consider the case that the expanding in RE is bounded. Theorem 3 gives the measure for the size, degree, and diameter of the target network for RE in this case:

Theorem 3. For $\{(n_f, d_f, k_f)\text{-}G_f, (n_u, d_u, k_u)\text{-}G_u\} \xrightarrow{\text{RE}^r} (n_r, d_r, k_r)\text{-}G_r$, the size, degree, and diameter of G_r follow the following equations:

1. If $1 \le r \le n_u$ and $\text{PSS}_{(s_k)}$ is taken,
   (a) $n_r \le n_u n_f^{n_u}$;
   (b) $d_r \le d_u + d_f$;
   (c) $k_r \le k_u(n_u + 1) + k_f n_u$.
2. If $d_f \le n_u$, $1 \le r \le \lfloor n_u/d_f \rfloor$ and $\text{PSS}_{(s_d)}$ is taken,
   (a) $n_r \le n_u n_f^{\lfloor n_u/d_f \rfloor}$;
   (b) $d_r \le d_u + 1$;
   (c) $k_r \le k_u(\lfloor n_u/d_f \rfloor + 1) + k_f n_u - n_u/2$.

Proof. The theorem is directly obtained from Theorem 2. ■

From Theorem 3 we can see that using two small networks, $(n_f, d_f, k_f)\text{-}G_f$ and $(n_u, d_u, k_u)\text{-}G_u$, as frame and unit, by RE we can get a large-size network G_r whose size is $O(n_f^{O(n_u)})$, degree is $O(\max\{d_u, d_f\})$, and diameter is $O((k_u + k_f)n_u)$. This result shows a way to construct a large network by simply taking two small networks with desired performance and properties and applying RE on them. Evidently, by Theorem 3, the target network produced by RE on the basis of two small networks can possess a lower cost than that of the initial ones; even the cost of the initial ones are already quite good.

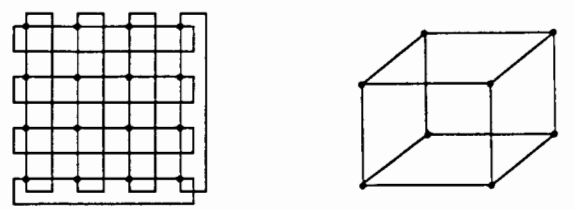
We define the *constructability* of a static network as the feasibility of the network to be constructed in arbitrary large size. Obviously, networks from RE can be constructed in arbitrary large size by choosing a proper frame and unit as well as setting a suitable number of expanding phases. Therefore, they have a high constructability.

Another important property of this kind of networks from RE is that the topologies of them are *symmetric*, which brings a series of advantages such as that the networks can be mathematically well described in a constructive way, that message routing can be realized in a simple and efficient way, and that without changing their inside structure, the networks can be easily extended into larger ones. In many cases, especially when the frame and the unit are regular, the target networks from RE are regular (we say a network is *regular* if all nodes of it have the same degree [2]).

4. TWO APPLICATIONS: THE $\mathcal{T}_r^{\Sigma}$ NETWORK AND $\mathcal{H}_r^{\Sigma}$ NETWORK

As showing two applications of the RE method, we take two kinds of well-known high-performance networks, the *torus* and *hypercube*, as shown in Figure 5, each as both the frame and the unit, and construct two target networks on the basis of them, the $\mathcal{T}_r^{\Sigma}$-network and the $\mathcal{H}_r^{\Sigma}$-network. Thereby, we exhibit the performance of the concrete target networks produced by RE and generate the cost-speedup of the $\mathcal{T}_r^{\Sigma}$-network and $\mathcal{H}_r^{\Sigma}$-network to conventional torus and hypercube, respectively.

It is known that a torus-sized n has a degree 4 and diameter $O(\sqrt{n})$ and that a hypercube sized n has a degree $\log n$ and diameter $\log n$ [1]. Denote a torus-sized n by $(n, 4, O(\sqrt{n}))\text{-}T$ and a hypercube-sized n by $(n, \log n, \log n)\text{-}H$. $\mathcal{T}_r^{\Sigma}$ and $\mathcal{H}_r^{\Sigma}$ are the target networks constructed by r phases RE on the basis of two toruses and two hypercubes as the frame and the unit, respectively. We have the following theorems exhibiting the performance of the target network from the two typical applications of RE:



(a) A 4 × 4 torus (b) A hypercube of 8 nodes

Fig. 5. The torus and hypercube structures.

Theorem 4. For $\{(n, 4, O(\sqrt{n}))\text{-}T, (n, 4, O(\sqrt{n}))\text{-}T\} \xrightarrow{\text{RE}^r} (n_r, d_r, k_r)\text{-}\mathcal{T}_r^{\Sigma}$, the size, degree, and diameter of $\mathcal{T}_r^{\Sigma}$ follow the following equations:

1. If $r = n$ and $\text{PSS}_{(s_k)}$ is taken,
   (a) $n_r = n^{n+1}$;
   (b) $d_r = 8$;
   (c) $k_r = O(\log^{3/2} n_r / \log^{3/2} n)$.
2. If $r = \lfloor n/4 \rfloor$ and $\text{PSS}_{(s_d)}$ is taken,
   (a) $n_r = n^{\lfloor n/4 \rfloor + 1}$;
   (b) $d_r = 5$;
   (c) $k_r = O(\log^{3/2} n_r / \log^{3/2} n)$.

Proof. The correctness of the theorem is evident by Theorem 3. ■

Corollary. For $\{(n, 4, O(\sqrt{n}))\text{-}T, (n, 4, O(\sqrt{n}))\text{-}T\} \xrightarrow{\text{RE}^r} (n_r, d_r, k_r)\text{-}\mathcal{T}_r^{\Sigma}$, $\lfloor n/4 \rfloor \le r \le n$, the cost-speedup of $\mathcal{T}_r^{\Sigma}$ to the same-size conventional torus, T_r, is $O[(\sqrt{n_r} \log^{3/2} \log n_r)(\log^{3/2} n_r)]$.

Proof. By Theorem 2, we know that n_r, d_r, and k_r will increase if n_u or n_f increases. So for $\lfloor n/4 \rfloor \le r \le n$, by Theorem 4, we have that

$$n^{\lfloor n/4 \rfloor + 1} \le n_r \le n^{n+1},$$

$$5 \le d_r \le 8,$$

$$k_r = O\left(\frac{\log^{3/2} n_r}{\log^{3/2} n}\right);$$

i.e.,

$$n_r = n^{O(n)}, \tag{12}$$

$$d_r = O(1), \tag{13}$$

$$k_r = O\left(\frac{\log^{3/2} n_r}{\log^{3/2} n}\right); \tag{14}$$

By (12) we have

$$\frac{\log n_r}{\log n} \ge \log^{1/2} n_r.$$

Thus,

$$\log n = O\left(\log \frac{\log n_r}{\log n}\right) \ge O(\log \log n_r).$$

Thereby, (14) becomes

$$k_r = O\left(\frac{\log^{3/2} n_r}{\log^{3/2} \log n_r}\right). \tag{15}$$

Here, we get the cost of G_r:

$$d_r \times k_r = O\left(\frac{\log^{3/2} n_r}{\log^{3/2} \log n_r}\right). \tag{16}$$

Since the cost of a torus with size n_r is $O(\sqrt{n_r})$, the corollary holds.

Figure 6 depicts an example of $\{(4, 2, 2)\text{-}T, (4, 2, 2)\text{-}T\} \xrightarrow{\text{RE}^2} (4^3, 3, 10)\text{-}\mathcal{T}_2^{\Sigma}$.

Theorem 5. For $\{(n, \log n, \log n)\text{-}H, (n, \log n, \log n)\text{-}H\} \xrightarrow{\text{RE}^r} (n_r, d_r, k_r)\text{-}\mathcal{H}_r^{\Sigma}$, if $r = O(n)$ and $\text{PSS}_{(s_k)}$ is taken, the size, degree, and diameter follow the following equations:

1. $n_r = n^{O(n)}$;
2. $dr = O(\log \log n_r)$;
3. $k_r = O(\log n_r)$.

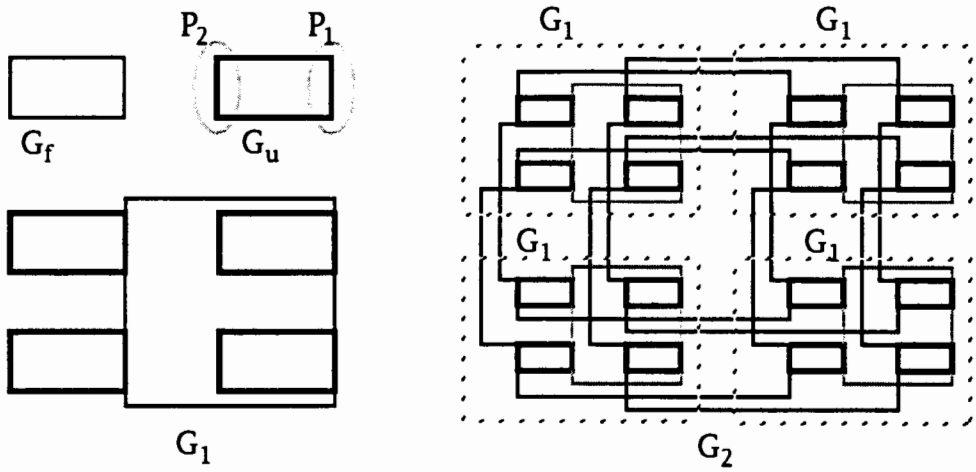


Fig. 6. An example of $\{(4, 2, 2)\text{-}T, (4, 2, 2)\text{-}T\} \xrightarrow{\text{RE}^2} (4^3, 3, 10)\text{-}\mathcal{T}_2^{\Sigma}$.

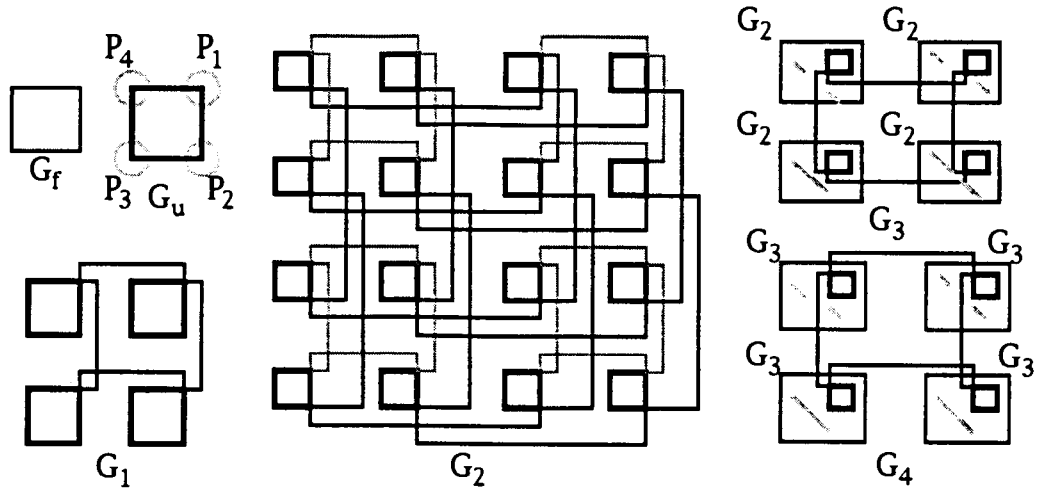


Fig. 7. An example of $\{(4, 2, 2)\text{-}H, (4, 2, 2)\text{-}H\} \xrightarrow{RE^4} (4^5, 4, 18)\text{-}\mathcal{H}_4^{\Sigma}$.

Proof. By Theorem 3, the equations for n_r and d_r evidently hold, and for k_r, since $n = O[(\log n_r)/(\log n)]$, we have

$$k_r = O(n \log n) = O(\log n_r).$$

Thus, the theorem holds. ■

Corollary. For $\{(n, \log n, \log n)\text{-}H, (n, \log n, \log n)\text{-}H\} \xrightarrow{RE^r} (n_r, d_r, k_r)\text{-}\mathcal{H}_r^{\Sigma}$, r = $O(n)$, the cost-speedup of $\mathcal{H}_r^{\Sigma}$ to the same-size conventional hypercube, H_r, is $O[(\log n_r)/(\log \log n_r)]$.

Proof. By Theorem 5, for $r = O(n)$, $\mathcal{H}_r^{\Sigma}$ has a cost $O(\log n_r \log \log n_r)$, where $n_r = n^{O(n)}$ is the size of G_r. Since a hypercube with size n_r has a cost $O(\log^2 n_r)$, the corollary holds. ■

Figure 7 depicts an example of $\{(4, 2, 2)\text{-}H, (4, 2, 2)\text{-}H\} \xrightarrow{RE^4} (4^5, 4, 18)\text{-}\mathcal{H}_4^{\Sigma}$.

5. THE ALGORITHM FOR AUTOMATIC NETWORK CONSTRUCTION

We are now going to give an algorithm for automatic network construction, which realizes $\{(n_f, d_f, k_f)\text{-}G_f, (n_u, d_u, k_u)\text{-}G_u\} \xrightarrow{RE^r} (n_r, d_r, k_r)\text{-}G_r$, where $\lambda n_u < r \leq (\lambda + 1)n_u$ and λ is an integer. In order to consider the general case, we choose $PSS_{(s_d)}$ in building our algorithm. The algorithm in the case of taking $PSS_{(s_k)}$ is obviously much simpler than the algorithm in the case of taking $PSS_{(s_d)}$, and it can be directly obtained from the first λn_u phases RE in our algorithm.

Let $A_{G_f}[1 \ldots n_f, 1 \ldots d_f]$ and $A_{G_u}[1 \ldots n_u, 1 \ldots d_u]$ be the *array representation* for G_f and G_u, where $A_{G_f}[i][1 \ldots d_f]$ and $A_{G_u}[j][1 \ldots d_u]$ represent the indices of all neighbors of node i in G_f and of node j in G_u, respectively. Let n_r nodes of G_r be indexed as $1, 2, \cdots, n_r$. Let the pivot sets in $PSS_{(s_d)}$ be indexed as $1, 2, \cdots, s_d$, where $s_d = n_u$ for the first λn_u phases RE and $s_d = r - \lambda n_u$ for the final $r - \lambda n_u$ phases RE, by Definition 4. We use an array, $A_{G_r}[1 \ldots n_r, 0 \ldots r + (\lambda + 1)d_f + 3]$, as the array representation for G_r. $A_{G_r}[i][j]$ $(1 \leq i \leq n_r, 0 \leq j \leq r)$ keeps the position (node index) in the G_f copy on which node i is placed during the jth-phase RE when $1 \leq j \leq r$ and the node index in the G_u copy of node i when $j = 0$. $A_{G_r}[i][r + 1]$ and $A_{G_r}[i][0]$ keep the index of the pivot set to which node i belongs in $PSS_{(s_d)}$ for the final $r - \lambda n_u$ phases RE, and for the first λn_u phases RE that is the same as the node index in G_u of node i, respectively. When node i is the first node (with the smallest index) in pivot set $A_{G_r}[i][0]$, $A_{G_r}[i][r + 2]$ is used as a pointer pointing the node in the pivot set to perform the next interunit connection, for the final $r - \lambda n_u$ phases RE. $A_{G_r}[i][r + 3 \ldots r + d_u + (\lambda + 1)d_f + 2]$ represents all neighbors of node i in G_r, which describes the topology of G_r, and $A_{G_r}[i][r + d_u + (\lambda + 1)d_f + 3]$ is used as a pointer pointing the first available position in $A_{G_r}[i][r + 3 \ldots r + d_u + (\lambda + 1)d_f + 2]$ for storing data during the computation for $A_{G_r}[i][r + 3 \ldots r + d_u + (\lambda + 1)d_f + 2]$, $1 \leq i \leq n_r$. Figure 8 describes the array $A_{G_r}[1 \ldots n_r, 0 \ldots r + d_u + (\lambda + 1)d_f + 3]$ as follows:

The array representation of G_r, $A_{G_r}[1 \ldots n_r, 0 \ldots r + d_u + (\lambda + 1)d_f + 3]$, can be distributed in the n_r nodes in G_r; thereby, node i is associated with the subarray $A_{G_r}[i][0 \ldots r + d_u + (\lambda + 1)d_f + 3]$, $1 \leq i \leq n_r$.

The algorithm for constructing G_r thus becomes the one for building the array representation of G_r. The basic idea of the algorithm can be caught as follows, by recalling the picture of the systematic proceeding of RE for G_r:

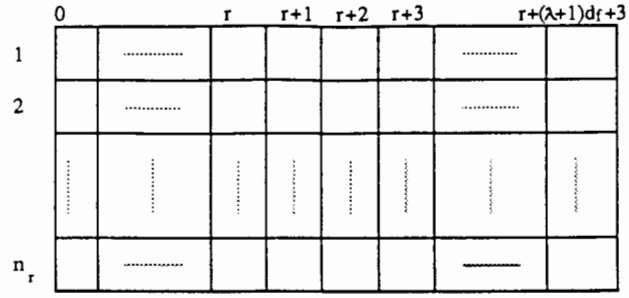


Fig. 8. The array representation of G_r.

1. For $1 \le i \le n_r$, $A_{G_r}[i][0 \,..\, r]$ can be easily computed in the way as illustrated in Figure 9.
2. For $1 \le i \le n_r$, $A_{G_r}[i][r+1]$ can be computed as follows: For the final $r - \lambda n_u$ phases RE, since the number of pivot sets in the pivot-set sequence is $r - \lambda n_u$ and the size of G_u is n_u, the size of each pivot set is therefore $n_{p_{set}} = \min\{\lfloor n_u/(r - \lambda n_u)\rfloor, d_f\}$. Thus for $1 \le i \le n_r$, node i belongs to pivot set $\lceil [1 + (i-1) \bmod(n_u)]/n_{p_{set}} \rceil$, by which we get $A_{G_r}[i][r+1]$.
3. $A_{G_r}[i][r+2]$ has an initial value 0, $1 \le i \le n_r$. New edges for the interunit connection are added to nodes in the index increasing order in each pivot set, during the final $r - \lambda n_u$ phases RE. When one new edge has been added to a node in a pivot set, the first node in the pivot set, assumed to be i_0, increases its $A_{G_r}[i_0][r+2]$ by 1, so as to point the next node in the pivot set to perform the interunit connection is $i_0 + A_{G_r}[i_0][r+2] \bmod(n_{p_{set}})$.
4. $A_{G_r}[i][r+3 \,..\, r + d_u + (\lambda+1)d_f + 2]$ consists of two parts: $A_{G_r}[i][r+3 \,..\, r + d_u + 2]$ representing the intraunit connection and $A_{G_r}[i][r + d_u + 3 \,..\, r + d_u + (\lambda+1)d_f + 2]$ representing interunit connection in r phases RE, $\lambda n_u < r \le (\lambda+1)n_u$, for $1 \le i \le n_r$. The intraunit connection can be directly obtained from A_{G_u}. The interunit connection depending on the topology of G_f is realized by adding new edges connecting nodes between corresponding pivot sets, which can be simply computed from A_{G_f} and the information on $PSS_{(s,j)}$ during the first λn_u phases RE and the final $r - \lambda n_u$ phases RE, respectively.

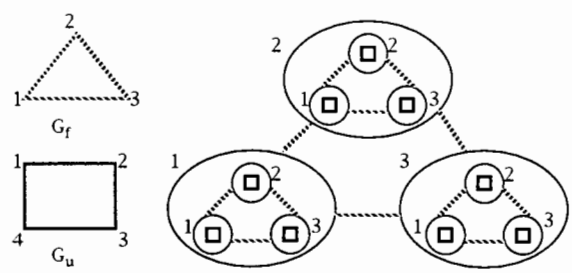


Fig. 9. The computation for $A_{G_r}[1 \,..\, n_r][0 \,..\, r + d_u + (\lambda+1)d_f + 3]$.

The algorithm for automatically constructing large interconnection networks of low cost by the RE method, namely, *RELLN*, is described as follows:

Algorithm *RELLN*$(A_{G_u}, A_{G_f}, r, n_r)$

{**Constructing the array representation of G_r, $A_{G_r}[1 \,..\, n_r, 0 \,..\, r + d_u + (\lambda+1)d_f + 3]$, for $\{(n_f, d_f, k_f)\text{-}G_f, (n_u, d_u, k_u)\text{-}G_u\} \xrightarrow{\text{RE}_r'} (n_r, d_r, k_r)\text{-}G_r$, where $n_r = n_u n_f^r$, $\lambda n_u < r \le (\lambda+1)n_u$ and λ is an integer.**}

for $i := 1$ **to** n_r **do**
  $A_{G_r}[i][0] := 1 + (i-1) \bmod(n_u)$;

{**Compute the position of node i in its relevant unit-copy (G_u).**}

  for $j := 0$ **to** $r - 1$ **do**
    $A_{G_r}[i][j+1] := 1 + \lceil (i-1)/n_j n_f^j \rceil \bmod(n_f)$;

{**Compute the position in frame G_f on which node i is placed during the $(j+1)$th-phase RE.**}

  $n_{p_{set}} := \min\{\lfloor n_u/(r - \lambda n_u)\rfloor, d_f\}$;

{**Compute the size of each pivot set, $n_{p_{set}}$, for the final $r - \lambda n_u$ phases RE.**}

  $A_{G_r}[i][r+1] := \lfloor 1 + (i-1) \bmod(n_u)/n_{p_{set}} \rfloor$;

{**Compute the index of the pivot set to which node i belongs during the final $r - \lambda n_u$ phases RE.**}

  $A_{G_r}[i][r+2) := 0$;
  $A_{G_r}[i][r + d_u + (\lambda+1)d_f + 3) := 0$;

{**Initialize the pointer in each pivot set for the final $r - \lambda n_u$ phases RE and the pointer to the available position for storing the information for interunit connection.**}

  for $j := r + 3$ **to** $r + d_u + 2$ **do**
    $A_{G_r}[i][j] := A_{G_u}[1 + (i-1) \bmod(n_u)][j - r - 2] + \lfloor i/n_u \rfloor n_u$;

{**Compute the intra-unit connections for node i.**}

  for $j := 1$ **to** r **do**
    for $\alpha := 1$ **to** n_f **do**
      for $\tau := 1$ **to** d_f **do**
        if $A_{G_f}[\alpha][\tau] > \alpha$ **then**
          $\beta := A_{G_f}[\alpha][\tau]$;

{*Find all different pairs of adjacent nodes, α and β, in G_f, for the interunit connection in the jth-phase RE.*}

if $j \le \lambda n_u$ **then**

{*Compute the interunit connections for node i during the first λn_u phases RE.*}

$pointer_i := 1 + (i - 1) \bmod (n_u)$;

{*$pointer_i$ points the pivot set $1 + (i - 1) \bmod (n_u)$ (containing only one node) in each unit of $G_{j-1}^{(\alpha)}$ and $G_{j-1}^{(\beta)}$.*}

while $pointer_i \le n_u n_f^{j-1}$ **do**
$pointer_j^{(\alpha)} := A_{G_r}[(\alpha - 1)n_u n_f^{j-1} + pointer_i][r + d_u + (\lambda + 1)d_f + 3)]$;
$pointer_j^{(\beta)} := A_{G_r}[(\beta - 1)n_u n_f^{j-1} + pointer_i][r + d_u + (\lambda + 1)d_f + 3)]$;

{*$G_{j-1}^{(\omega)}$ has a size $n_u n_f^{j-1}$ and contains nodes from $(\omega - 1)n_u n_f^{j-1} + 1$ to $\omega n_u n_f^{j-1}$, $pointer_j^{(\omega)}$ points the available position in $A_{G_r}[(\omega - 1)n_u n_f^{j-1} + pointer_i][r + d_u + 3 \,..\, r + d_u + (\lambda + 1)d_f + 2]$, $\omega = \alpha, \beta$.*}

$A_{G_r}[(\alpha - 1)n_u n_f^{j-1} + pointer_i][r + d_u + 3 + pointer_j^{(\alpha)}] :=$
$(\beta - 1)n_u n_f^{j-1} + pointer_i$;
$A_{G_r}[(\beta - 1)n_u n_f^{j-1} + pointer_i][r + d_u + 3 + pointer_j^{(\beta)}] :=$
$(\alpha - 1)n_u n_f^{j-1} + pointer_i$;
$A_{G_r}[(\alpha - 1)n_u n_f^{j-1} + pointer_i][r + d_u + (\lambda + 1)d_f + 3] := pointer_j^{(\alpha)} + 1$;
$A_{G_r}[(\beta - 1)n_u n_f^{j-1} + pointer_i][r + d_u + (\lambda + 1)d_f + 3] := pointer_j^{(\beta)} + 1$;
$pointer_i := pointer_i + n_u$;

{*$pointer_i$ moves to the corresponding position in the next unit in $G_{j-1}^{(\alpha)}$ and $G_{j-1}^{(\beta)}$.*}

if $j > \lambda n_u$ **then**

{*Compute the inter-unit connections for node i during the final $r - \lambda n_u$ phases RE.*}

$n_{p_{set}} := \min\{\lceil n_u/(r - \lambda n_u) \rceil, d_f\}$;
$head_i := ((i - 1) \bmod (n_u))n_{p_{set}} + 1$;

{*$n_{p_{set}}$ is the size of each pivot set in the final $r - \lambda n_u$ phases RE, $head_i$ points the first element of pivot set $1 + (i - 1) \bmod (n_u)$ in each unit of $G_{j-1}^{(\alpha)}$ and $G_{j-1}^{(\beta)}$.*}

while $head_i \le n_u n_f^{j-1}$ **do**
$pointer_{j,\alpha} := A_{G_r}[(\alpha - 1)n_u n_f^{j-1} + head_i][r + 2]$;
$pointer_{j,\beta} := A_{G_r}[(\beta - 1)n_u n_f^{j-1} + head_i][r + 2]$;

{*$pointer_{j,\omega}$ points the node in pivot set $1 + (j - 1) \bmod (n_u)$ in each unit of G_{j-1}^{ω} for the coming interunit connection, $\omega = \alpha, \beta$.*}

$pointer_j^{(\alpha)} := A_{G_r}[(\alpha - 1)n_u n_f^{j-1} + head_i + pointer_{j,\alpha}][r + d_u + (\lambda + 1)d_f + 3]$;
$pointer_j^{(\beta)} := A_{G_r}[(\beta - 1)n_u n_f^{j-1} + head_i + pointer_{j,\beta}][r + d_u + (\lambda + 1)d_f + 3]$;
$A_{G_r}[(\alpha - 1)n_u n_f^{j-1} + head_i + pointer_{j,\alpha}][r + d_u + 3 + pointer_j^{(\alpha)}] :=$
$(\beta - 1)n_u n_f^{j-1} + head_i + pointer_{j,\beta}$;
$A_{G_r}[(\beta - 1)n_u n_f^{j-1} + head_i + pointer_{j,\beta}][r + d_u + 3 + pointer_j^{(\beta)}] :=$
$(\alpha - 1)n_u n_f^{j-1} + head_i + pointer_{j,\alpha}$;
$A_{G_r}[(\alpha - 1)n_u n_f^{j-1} + head_i + pointer_{j,\alpha}][r + d_u + (\lambda + 1)d_f + 3 :=$
$pointer_j^{(\alpha)} + 1$;
$A_{G_r}[(\beta - 1)n_u n_f^{j-1} + head_i + pointer_{j,\beta}][r + d_u + (\lambda + 1)d_f + 3 :=$
$pointer_j^{(\beta)} + 1$;
$A_{G_r}[(\alpha - 1)n_u n_f^{j-1} + head_i][r + 2] := (pointer_{j,\alpha} + 1) \bmod (n_{p_{set}})$;
$A_{G_r}[(\beta - 1)n_u n_f^{j-1} + head_i][r + 2] := (pointer_{j,\beta} + 1) \bmod (n_{p_{set}})$;

{*Adjust $pointer_{j,\omega}$ to the node in pivot set $1 + (j - 1) \bmod (n_u)$ of each unit in G_{j-1}^{ω} for the next interunit connection, $\omega = \alpha, \beta$.*}

$head_i := head_i + n_u$;

{*$head_i$ moves to the corresponding position at the next unit in $G_{j-1}^{(\alpha)}$ and $G_{j-1}^{(\beta)}$.*}

end.

The correctness of the algorithm can be easily seen from the verifications of the comments interspersed with the program text. The time complexity of the algorithm, t_{n_r}, can be evidently expressed as follows:

$$t_{n_r} = n_r(O(1) + O(r + d_u) + O(rd_f n_f^r)) = O\left(\max\left\{\frac{d_f}{n_u}, \frac{d_u}{r}\right\} rn_r^2\right). \quad (17)$$

When d_u, $d_f = O(1)$, $n_u = O(n_f)$ and $r = O(n_u)$, the above equation becomes

$$t_{n_r} = O(n_r^2). \quad (18)$$

Obviously, Eq. (18) will also hold when $d_f/n_u = O(d_u/r)$ and $d_u = O(1)$.

6. ROUTING IN THE NETWORKS CONSTRUCTED BY RE

From the RE method and the algorithm RELLN as described above, we know that networks constructed by RE have a very symmetric topology. The indices of the nodes of each pivot set performing the interunit connection in this kind of network are implicitly kept by the pivot index and the pivot size; therefore, each of those nodes is directly accessible to the other nodes by only knowing the pivot set index and size. These features make the message routing in those networks to be possibly realized in a simple and efficient way.

Let A_{G_r} be the representation array of G_r built by RELLN for $\{(n_f, d_f, k_f)\text{-}G_f, (n_u, d_u, k_u)\text{-}G_u\} \xrightarrow{\text{RE}_r} (n_r, d_r, k_r)\text{-}G_r$, where $\lambda n_u < r \leq (\lambda + 1)n_u$ and λ is an integer. $\text{SEND}_{G_r}(x, y, msg, A_{G_r})$ denotes sending message *msg* from node x to node y in G_r, where x and y are adjacent. Since the frame, G_f, and the unit, G_u, normally have a small size, routing within G_f and G_u therefore is usually not difficult and we suppose that this has already been realized by existing techniques (if we take some standard regular networks, for instance, torus or hypercube, as G_f and G_u, there are always standard routing algorithms associated with them). Assume that message routing within G_f and G_u are realized by two procedures: ROUT_f and ROUT_u. Let $\text{ROUT}_f(x, y, msg, first, last, \Delta, \text{RETURN}(path))$ and $\text{ROUT}_u(x, y, msg, first, last, \Delta)$ realize sending *msg* from node s to node t in a G_f-copy and G_u-copy whose nodes are indexed from *first* to *last* by an index-step Δ (the index-difference between two consequent indices), respectively, where "RETURN(*path*)" means returning the path information in which the message has been routed. By Definition 4 we know that all nodes of each pivot set in the unit are connected, so message passing between any pair of nodes in a pivot set can be realized through internode communication inside the region of the pivot set. Because the number of nodes of each pivot set is never greater than d_f, routing within the region of a pivot set is trivial and obviously easier than routing within G_u. Let $\text{ROUT}_{P_m}(x, y, msg, A_{R(P_m)}, first, last)$ be the procedure sending *msg* from node x to node y in $R(P_m)$, the region of pivot set P_m, whose representation array is $A_{R(P_m)}$ and nodes are indexed from *first* to *last*, consequently, $1 \leq m \leq s_d$. Clearly, in G_m containing n_f G_{m-1}-copies: $G_{m-1}^{(1)}, \ldots, G_{m-1}^{(n_f)}$, for each unit copy in $G_{m-1}^{(j)}$ $(1 \leq j \leq n_f)$, all corresponding unit copies in n_f G_{m-1} copies are connected in a topology of G_f via pivot set P_m. Assume that $G_u^{(1,\tilde{i})}, \ldots, G_u^{(n_f,\tilde{i})}$ are n_f corresponding unit copies to $G_u^{(s,\tilde{i})}$ $(1 \leq s \leq n_f)$ containing node i, where $1 \leq \tilde{i} \leq n_f^{m-1}$ (the number of unit copies in a G_{m-1} copy), $G_u^{(j,\tilde{i})} \subset G_{m-1}^{(j)}$ and $P_m^{(j,\tilde{i})}$ is the pivot set P_m of $G_u^{(j,\tilde{i})}$ for $1 \leq j \leq n_f$. Message routing between nodes in $P_m^{(s,\tilde{i})}$ and $P_m^{(t,\tilde{i})}$ $(s \neq t)$ is performed by alternatively routing message intra-$R(P_m)$ and inter-$R(P_m)$, as shown in Figure 10. By combining $\text{ROUT}_f(s, t, msg, A_{G_f}, first, last, 1, \text{RETURN}(path))$ and $\text{Rout}_{P_m}(x, y, msg, A_{R(P_m)}, first, last)$, we can easily obtain a procedure, $\text{ROUT}_{f(P_m)}(x, t.first, msg, A_{R(P_m)}, A_{G_f}, A_{G_r}, first, \Delta, \text{RETURN}(tt))$, which realizes sending *msg* from node x in pivot set $P_m^{(s,x)}$ (of the unit copy $G_u^{(s,\tilde{x})}$ in $G_{m-1}^{(s)}$) to a node in another pivot set $P_m^{(t,\tilde{x})}$ (of the corresponding unit copy in $G_{m-1}^{(t)}$) whose first node is $t.first$, where $1 \leq s = \lceil x/\Delta \rceil$, $t = \lceil t.first/\Delta \rceil \leq n_f$, $\Delta = n_{m-1}$ (the number of nodes in G_{m-1}) and *first* is the first node of $P_m^{(1,\tilde{x})}$ $(1 \leq m \leq s_d)$, and, finally, returns the index of the node *tt* in $P_m^{(t,\tilde{x})}$, in which *msg* has arrived.

Procedure $\text{ROUT}_{f(P_m)}(x, t.first, msg, A_{P_m}, A_{G_f}, A_{G_r}, first, \Delta, \text{RETURN}(tt))$ works in the following way:

1. Call $\text{ROUT}_f(s, t, msg, A_{G_f}, 1, n_f, 1, \text{RETURN}(path))$ to obtain an intra-G_f shortest path with a length at most k_f from node s to node t, where $s = \lceil x/\Delta \rceil$, $t = \lceil t.first/\Delta \rceil$ and $\Delta = n_{m-1}$: $s = \tau_0 \rightarrow \tau_1 \rightarrow \cdots\cdots \rightarrow \tau_l = t$, $1 \leq \tau_j \leq n_f$ and $1 \leq j \leq l \leq k_f$.

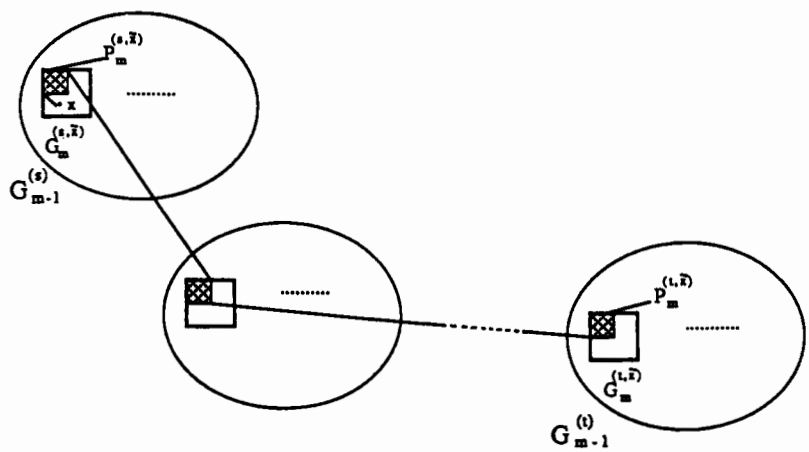


Fig. 10

Because $R(P_m^{(1,\bar{x})}), \cdots, R(P_m^{(n_f,\bar{x})})$ are connected in a topology of G_f, we know the inter-$R(P_m)$ path from $R(P_m^{(s,\bar{x})})$ to $R(P_m^{(t,\bar{x})})$ is $R(P_m^{(s,\bar{x})}) = R(P_m^{(\tau_0,\bar{x})}) \rightarrow R(P_m^{(\tau_1,\bar{x})}) \rightarrow \cdots \rightarrow R(P_m^{(\tau_l,\bar{x})}) = R(P_m^{(t,\bar{x})})$, where the first node of $P_m^{(\tau_j,\bar{x})}$ is $first.\tau_j = first + \Delta(\tau_j - 1)$, $1 \le \tau_j \le n_f$ and $1 \le j \le l \le k_f$.

2. Let $x'_0 = x$. For each τ_j, $0 \le j < l \le k_f$,

   (a) Find a concrete node, x_j in $P_m^{(\tau_j,\bar{x})}$, where there is an edge (added by the $\bar{m}$th-phase RE) ($m = 1 + (\bar{m} - 1) \bmod (n_u)$) connecting to [a node (assumed to be x'_{j+1}) of] $P_m^{(\tau_{j+1},\bar{x})}$ by looking up $A_{G_r}[e][r + d_u + 3 \,..\, r + d_u + \bar{m} + 2]$ for $first.\tau_j \le e \le first.\tau_j + |P_m| - 1$;

   (b) Call $\mathrm{ROUT}_{P_m}(x'_j, x_j, msg, A_{R(P_m)}, first.\tau_j, first.\tau_j + |P_m| - 1)$, i.e., send msg in an intra-$R(P_m^{(\tau_j,\bar{x})})$ path with a length at most $|P_m| \le d_f$ from x'_j to x_j;

   (c) $\mathrm{SEND}_{G_r}(x_j, x'_{j+1}, msg, A_{G_r})$, i.e., send msg from x_j to x'_{j+1} in G_r.

3. Put the value of x'_{j+1}, i.e., the index of a node in $P_m^{(t,\bar{x})}$ in which msg has arrived, to variable tt and return.

Message routing in G_r is realized by a kind of multi-phase routing from the highest phase ($\le r$) to the lowest phase (0) according to the indices (addresses) of the source node and destination node, based on the information kept in the representation array A_{G_r}. The basic idea and sketch of our algorithm for message routing between any pair of nodes indexed x and y, where x sends a message to y, in G_r is described as follows:

1. Find the minimum value of m, such that both node x and y reside in two different G_{m-1} copies of a G_m copy ($0 \le m \le r$, $G_0 = G_u$), where the G_m copy constructed by m phases RE on G_f and G_u is named the *most significant subnetwork of* G_r. This can be realized by looking up the information in A_{G_r} and finding the highest phase of RE by which node x and y have been placed onto different positions in G_f. Thus, message routing from node x to y will be carried out within a single G_m copy.
2. Now node x and y belong to two different G_{m-1}'s: $G_{m-1}^{(s)}$ and $G_{m-1}^{(t)}$, $1 \le s, t \le n_f$, in the most significant subnetwork G_m. Message routing from node x to y can be realized as the same way as shown in the proof of Lemma 3 of Theorem 2 (for the diameter of G_r):

   (a) Send the message from node x to a node, p, in the pivot set $P_m^{(s,\bar{x})}$ of the unit-copy $G_u^{(s,\bar{x})}$ containing node x, which can be accomplished by routing along with the corresponding path from $1 + (x - 1) \bmod (n_u)$ to $1 + (p - 1) \bmod (n_u)$ produced by calling the procedure ROUT_u. For $m \le \lfloor r/n_u \rfloor n_u$, i.e., the mth-phase RE belonging to the first $\lfloor r/n_u \rfloor n_u$ phases of the r phases ($\lfloor r/n_u \rfloor n_u \le r < (\lfloor r/n_u \rfloor + 1)n_u$), there is only one node in $P_m^{(s,\bar{x})}$, so just send the message to this node. Otherwise, in the case that $P_m^{(s,\bar{x})}$ contains more than one node, use the pointer stored in $A_{G_r}[head][r + 2]$ and send the message to the node pointed by the pointer and then adjust the pointer to the next node in the pivot set to receive the next message, so as to make all nodes in the pivot set to be used for message passing as evenly as possible. Note that instead of sending the message to the node pointed by the pointer, of course, we can also set p as the node that initiates a shortest path from $P_m^{(s,\bar{x})}$ to $P_m^{(t,\bar{x})}$ obtained by calling the procedure ROUT_f in order to eliminate intra-$R(P_m^{(s,)})$ message passing, as shown in the proof of Lemma 3 of Theorem 2.

   (b) For $m \le \lfloor r/n_u \rfloor n_u$, send the message from node p of $P_m^{(s,\bar{x})}$ in $G_{m-1}^{(s)}$ to the node q of the corresponding $P_m^{(t,\bar{x})}$ in $G_{m-1}^{(t)}$ that is generated by a series of inter-$R(P_m)$ message passing between different G_{m-1} from $P_m^{(s,\bar{x})}$ in $G_{m-1}^{(s)}$ to $P_m^{(t,\bar{x})}$ in $G_{m-1}^{(t)}$ along with the corresponding path from node s to node t produced by calling the procedure ROUT_f. Otherwise, for the case that pivot set m contains more than one node, send the message from node p of $P_m^{(s,\bar{x})}$ in $G_{m-1}^{(s)}$ to a node of $P_m^{(t,\bar{x})}$ in $G_{m-1}^{(t)}$ that the message can reach by calling procedure $\mathrm{ROUT}_{f(P_x)}$. The index of the node, tt, in $G_{m-1}^{(t)}$ that has received the message from p is obtained from $\mathrm{ROUT}_{f(P_x)}$.

   (c) Send the message from either node q if $m \le \lfloor r/n_u \rfloor n_u$ or node tt otherwise to y in $G_{m-1}^{(t)}$. If $m - 1 > 0$, recursively call the above routing procedure within the decomposed subnetwork $G_{m-1}^{(t)}$. Otherwise, the routing within a G_u can be simply accomplished along with the corresponding path produced by calling the procedure ROUT_u, and, therefore, the routing from node x to y in G_r is finished.

Our routing algorithm, ROUTRCN, is presented as follows:

Algorithm *ROUTRCN*(x, y, msg, A_{G_r})

{**Rout message msg from node x to y in network G_r with representation array A_{G_r}**}

$m := r$;
while $(A_{G_r}[x][m] = A_{G_r}[y][m]) \wedge (m > 0)$ **do**
$m := m - 1$;

{*Find the most significant subnetwork G_m in G_r.*}

if $m \leq \lfloor r/n_u \rfloor n_u$ **then**

{*The m-th phase RE belongs to the first $\lfloor r/n_u \rfloor n_u$ phases of the r phases.*}

$ss := x$;
$tt := (x - (x - 1)\text{mod}(n_u)) + (m - 1)\text{mod}(n_u)$;
$first := x - (x - 1)\text{mod}(n_u)$;
$last := first + n_u - 1$;
$\text{ROUT}_u(ss, tt, msg, A_{G_u}, first, last, 1)$;

{* Send msg from node ss to the node, tt, in pivot set $P_m^{(s,\vec{x})}$ of the unit copy containing node x.*}

$ss := tt$;
$tt := (y - (y - 1)\text{mod}(n_u)) + (m - 1)\text{mod}(n_u)$;
$first := (first - 1)\text{mod}(n_u n_f^{m-1}) + (m - 1)\text{mod}(n_u) + 1$;
$last := first + n_u n_f^{m-1}(n_f - 1)$;
$\text{ROUT}_f(ss, tt, msg, A_{G_f}, first, last, n_u n_f^{m-1}, RETURN(path))$;

{* Send msg from node in $P_m^{(s,\vec{x})}$ of the unit copy containing node x to the node, tt, in $P_m^{(t,\vec{x})}$ of the corresponding unit copy in $G_{m-1}^{(t)}$ containing node y.*}

$\text{ROUTRCN}(tt, y, msg, A_{G_r})$;

{*Recursively route msg from the node tt to y within the decomposed subnetwork $G_{m-1}^{(t)}$.*}

if $m > \lfloor r/n_u \rfloor n_u$ **then**

{*The m-th phase RE belongs to the final $r - \lfloor r/n_u \rfloor n_u$ phases of the r phases.*}

$ss := x$;
$n_{p_{set}} := \min\{[n_u/(r - \lfloor r/n_u \rfloor n_u)], d_f\}$;
$head_m^{(s,\vec{x})} := (i - (i - 1)\text{mod}(n_u)) + (m - 1)\text{mod}(n_u) n_{p_{set}}$;
$tt := head_m^{(s,\vec{x})} + A_{G_r}[head_m^{(s,\vec{x})}][r + 2]$;
$A_{G_r}[head_m^{(s,\vec{x})}][r + 2] := (A_{G_r}[head_m^{(s,\vec{x})}][r + 2] + 1)\text{mod}(n_{p_{set}})$;

{*$n_{p_{set}}$ is the size of each pivot set, $head_m^{(s,\vec{x})}$ is the index of the first node (smallest index) in pivot set $P_m^{(s,\vec{x})}$ of the unit copy containing node i, $A_{G_r}[head_m][r + 2]$ is a pointer.*}

$first := x - (x - 1)\text{mod}(n_u)$;
$last := first + n_u - 1$;
$\text{ROUT}_u(ss, tt, msg, A_{G_u}, first, last, 1)$;

{*Send msg from node ss to the node, tt, in P_m of the unit copy containing node x.*}

$ss := tt$;
$head_{G_{m-1}^{(t)}} := \lfloor y/n_u n_f^{m-1} \rfloor n_u n_f^{m-1} + 1$;
$t.first := head_{G_{m-1}^{(t)} + head_m^{(s,\vec{x})}} - 1$;

{*$head_{G_{m-1}^{(t)}}$ is the index of the first node (smallest index) in $G_{m-1}^{(t)}$, t.first is the index of the first node of pivot set $P_m^{(t,\vec{x})}$ of the unit copy in $G_{m-1}^{(t)}$ with the same position as the unit copy containing node x in $G_{m-1}^{(s)}$.*}

$first := (first - 1)\text{mod}(n_u n_f^{m-1}) + (m - 1)\text{mod}(n_u) + 1$;
$last := first + n_u n_f^{m-1}(n_f - 1)$;
$\text{ROUT}_{f(P_m)}(ss, t.first, msg, A_{R(P_m)}, A_{G_f}, A_{G_r}, first, n_u n_f^{m-1}, RETURN(tt))$;

{*Send msg from node ss of $P_m^{(s,\vec{x})}$ of the unit copy ($G_u^{(s,\vec{x})}$) containing node x to $P_m^{(t,\vec{x})}$ whose first node is t.first of the corresponding unit copy in $G_{m-1}^{(t)}(G_u^{(t,\vec{x})})$, where $R(P_m^{(1,\vec{x})}), \cdots, R(P_m^{(n_f,\vec{x})})$, i.e., all $R(P_m)$ of the corresponding unit copies (to the unit copy containing node x) in $G_{m-1}^{(1)}, \cdots, G_{m-1}^{(n_f)}$, are connected in a topology of G_f, and first is the first node of $P_m^{(1,\vec{x})}$, and return the index of the node, tt, in which msg has arrived.*}

$\text{ROUTRCN}(tt, y, msg, A_{G_r})$

{*Recursively route msg from the node tt to y within the decomposed subnetwork $G_{m-1}^{(t)}$.*}

end.

The correctness of the algorithm can be easily seen from the comments interspersed with the program text. The time complexity of the algorithm, i.e., the maximum delay of message passing between any pair of nodes in G_r, is never greater than $O((k_u + k_f)d_f r)$, which can be directly derived from Lemma 3 of Theorem 2.

7. CONCLUDING REMARKS

Motivated by solving the problem of automatically and systematically constructing arbitrary large intercon-

nection networks with high performance, in this paper we have proposed a novel method, *recursive expansion*, or RE for short, for constructing large networks with high performance. The basic idea of this method is based on the observation that by taking two small networks we can construct a network of much larger size than that of them by just simply replacing each node in the frame with a unit and each edge in the frame with a set of interunit connections. Thus, taking the result network as a new unit and recursively expanding the unit on the frame can lead to a network of an arbitrary large size. By carefully choosing the frame and the unit, we can make the target network possess the desired properties and high performance. The RE method has the following properties:

- RE can produce arbitrary large networks of arbitrary topologies in a systematic manner, which provides a methodology for automatic network design.
- Networks constructed by RE have a symmetric topology, which brings a series of advantages of this kind of networks such as constructive description, high extendability, and simple message-routing.
- Networks constructed by RE have a very low cost. As two typical applications of RE, the $\mathcal{T}_r^{\Sigma}$ network on the basis of two small *torus* and the $\mathcal{H}_r^{\Sigma}$ network on two small *hypercubes* have a cost $O[(\log^{3/2} n)/(\log^{3/2} \log n)]$ (degree $O(1)$) and $O(\log n \log \log n)$ (degree $O(\log \log n)$), respectively, where n is the size of the networks.
- Networks constructed by RE possess good *extendability*. A network can be easily extended without changing its structure by just taking it as a single unit and applying RE again.
- Networks can be constructed with respect to minimizing either the degree or the diameter, depending on the design requirements, by taking the corresponding pivot-set sequence.

The proposed RE method constructs networks of low cost with respect to the product of degree and diameter. If the number of edges is also a measurement for cost, since the degree implies the number of edges required for connecting the given set of nodes in a network of regular topology and networks constructed by the RE method have a highly regular topology unless the underlying frame and unit are extremely irregular, the RE method will, in most cases, also produce networks of low cost in this sense.

Based on the RE method, we have described an algorithm for automatically constructing networks of arbitrary large size with high performance. For constructing a network of size n_r through r phases RE on the basis of a frame of degree d_f and a unit of size n_u and degree d_u, the algorithm has a time complexity $O((\max\{d_f/n_u, d_u/r\})rn_r^2)$. Furthermore, we have presented a routing algorithm for message routing in the networks constructed by RE. Message routing from a source to a destination in a network constructed by r phases RE is realized in a multiphase point-to-point routing manner from the highest phase within the network to the lowest phase within a unit copy, stepwise approaching the destination. Unlike usual point-to-point routing schemes, our routing here does not need to keep a global routing table at each node, which is quite space-consuming, because in the networks constructed by RE the interunit connection during each phase RE is realized by regularly connecting nodes among pivot sets with a fixed index. For routing in a network constructed by r phases RE on the basis of a frame of degree d_f and diameter k_f and a unit of diameter k_u, the algorithm has a time complexity $O((k_u + k_f)d_f r)$.

For ease of analysis, in the above RE, we assume that the topology of the frame G_f is fixed, i.e., all phases RE use the same G_f. This is, however, not necessary in the general case. Our RE method can also be applied in the case that the frame is variable, i.e., different phases RE can use different frames, by just simply specifying a sequence of frames: $G_{f_1}, \cdots, G_{f_r}$ as input data before the execution of RE and substituting G_f with G_{f_i} in the i-th phase RE during the execution, $1 \le i \le r$. Furthermore, the partitioning of the pivot sets (in the node set of the unit) for interunit connection in the general case can also be variable, depending on the property requirement of the target network.

The authors wish to thank Aimo Törn for his discussion on the manuscript of the paper. The authors also wish to thank an anonymous referee for the comments and suggestions on the paper.

REFERENCES

[1] G. S. Almasi and A. Gottlieb, *Highly Parallel Computing*. Benjamin/Cummings, (1988).

[2] G. A. Anderson and E. D. Jensen, Computer interconnection structure: Taxonomy, characteristics, and examples. *ACM Comput. Surv.* 7(4) (1975) 197–213.

[3] L. W. Beineke and R. J. Wilson, *Selected Topics in Graph Theory*, Vol. 2, Academic Press, New York (1983).

[4] J.-C. Bermond and C. Delorme, Strategies for intercon-

nection networks: Some methods from graph theory. *J. Parallel Distributed Comput.* **3** (1986) 433–449.

[5] T. Feng, A survey of interconnection networks. *Computer* **14**(12) (1981) 12–27.

[6] W. D. Hillis, *The Connection Machine*, Cambridge, MA (1985).

[7] L. D. Wittie, Communication structures for large networks of microcomputers. *IEEE Trans. Comput.* **C-30**(4) (1981) 264–273.

[8] N. S. Woo and A. Agrawala, A symmetric tree structure interconnection network and its message traffic. *IEEE Trans. Comput.* **C-34**(8) (1985) 765–769.