A Conflict-Free Routing Scheme on Multistage Interconnection Networks

WOEI LIN, MEMBER, IEEE, TSANG-LING SHEU, STUDENT MEMBER, IEEE, CHITA R. DAS, MEMBER, IEEE,
TSE-YUN FENG, FELLOW, IEEE, AND CHUAN-LIN WU, SENIOR MEMBER, IEEE

Abstract—**In this paper, we present a conflict-free routing scheme for a class of parallel and distributed computing systems. The core of the proposed routing scheme is a quadtree communication structure. The quadtree structure suggests a general approach to mapping a class of parallel algorithms with intensive communication requirements for selecting data from many different sources and distributing data from a single source. By properly merging messages and efficiently replicating data, the quadtree structure can complete required communications in $O(\log_4 M)$ parallel steps, where M is the network size. We also consider contraction and stretch of the quadtree structure, while retaining its conflict-free property. Contracted and stretched tree structures allow us to adaptively balance off computation and communication requirements for various parallel algorithms.**

Index Terms—**Interprocessor communication, multistage interconnection networks, network contentions, network routing, parallel algorithms.**

I. Introduction

SELECTING data from different sources and distributing data to various destinations are two essential operations in many parallel and distributed computations. Matrix inversion with pivots [1] and linear programming algorithms with simplex method [2] are two typical examples which heavily make use of the two operations: pivot elements are iteratively selected from and transmitted to many data elements. These two operations are intrinsically communication intensive especially in a shared-memory multiprocessor environment. In this paper, we present an efficient, conflict-free routing scheme for fast data selection and distribution on a class of multistage interconnection networks (MIN's).

It is widely known that the MIN offers better throughput and flexibility yet with less hardware complexity when used for large-scale parallel and distributed computing systems. Wu and Feng [3] have shown that the topological equivalence of a class of multistage interconnection networks, which include data manipulator [4], baseline [3], omega [5], Flip [6], SW-banyan ($s = f = 2$) [7], indirect binary n-cube [8], and shuffle exchange network [9]. The MIN has been widely used as an interconnection scheme in multiprocessor systems. Examples include BBN Butterfly Parallel Processor [10], STARAN [11], IBM RP3 [12], NYU Ultracomputer [13], PASM [14], etc. Many multiprocessor systems use a network constructed with 2 × 2 switching elements to interconnect processor nodes. However, the advent of VLSI technology now makes it possible to implement single-chip switching elements of larger size. This becomes the current trend and promises better performance and more flexible permuting capability [15], [16]. As an example, the BBN Butterfly Parallel Processor uses 4 × 4 switching elements in the communication network. Our proposed routing scheme is primarily designed for MIN's with 4 × 4 switches. However, it can be applied to networks with different sizes of switching elements, such as 2 × 2, 8 × 8, etc.

Contentions in the MIN, if not carefully controlled, may cause serious interprocessor communication delay and consequently degrade the system performance. There are two major factors that contribute to interprocessor communication delay in the MIN-based multiprocessor system. These two factors are 1) contentions for shared memories and 2) conflicts in communication links. Performance analysis of the Butterfly Parallel Processor has been presented in [17] and [18]. These studies point out that if contention problems in shared memories and communication links become dominant, the speedup curve levels off rapidly as more processors are added. LeBlanc also examines the effect of memory and switch contention by adding extra memories and extra switches in the system network [19]. He concludes that an implementation based on very efficient communication (e.g., shared memory) may *perform worse* than that based on a less efficient mechanism (e.g., distributed memory), if the former implementation has too much communication overhead due to memory and switch contention. Hence, reduction of the network contentions is a critical consideration for achieving high performance in the MIN-based multiprocessor environment.

In this paper, we propose a conflict-free routing scheme for reducing the two contention problems to the minimum. The proposed scheme does not require hardware augmentation in communication networks. Contention costs both for shared memories and communication links can be effectively minimized by using the conflict-free routing scheme. The routing scheme relies on the construction of a quadtree structure on the MIN. The quadtree structure can be, at will, contracted and stretched to suit the needs of various parallel algorithm applications, while its conflict-free property is still retained. The proposed routing scheme is distributed in nature; it allows

---

Manuscript received September 29, 1988; revised April 16, 1989.

W. Lin is with the Department of Electrical and Computer Engineering, University of New Mexico, Albuquerque, NM 87131.

T.-L. Sheu, C. R. Das, and T.-Y. Feng are with the Computer Engineering Program, Department of Electrical Engineering, The Pennsylvania State University, University Park, PA 16802.

C.-L. Wu is with the Department of Electrical Engineering, University of Texas at Austin, Austin, TX 78712.

IEEE Log Number 8928533.

0018-9340/89/0800-1086$01.00

data messages to be disseminated or combined under distributed control.

The rest of this paper is organized as follows. In Section II, we introduce the configuration and routing of a class of MIN's. In Section III, the conflict-free routing scheme is presented. Procedures of constructing communication structures for conflict-free routing are described. In Section IV, methods for contracting and stretching the quadtree structure are presented. The effectiveness of the quadtree communication structure is analyzed in Section V. Section VI gives a concluding remark.

II. Network Characteristics and Routing

A. Network Characteristics

A multistage interconnection network consists of stages of switching elements and communication links arranged in certain permutation patterns. The size of switching elements is generally $2^n \times 2^n$, where n is a positive integer. The class of networks with 2^m inputs and 2^m outputs, where m is a multiple of n, have the following characteristics:

1) number of stages = $(m/n) + 1$;

2) number of switching elements in each stage (except the last stage) = 2^{m-n};

3) total number of switching elements = $(m/n) \times 2^{m-n}$;

4) regular, symmetric permutation between adjacent stages.

The stages are numbered from $i = 0$ to m/n. Let $M = 2^m$ represent the number of processors. The network has an advantage that the total number of switching elements is $O(M \log M)$. The two major network parameters m and n are hereafter referred to as *network dimension* and *switch dimension,* respectively. In the subsequent sections, we choose the baseline network as the primary network to proceed the discussion, because of its universality that it is topologically equivalent to many other multistage interconnection networks. The BBN Butterfly switching network, shown in Fig. 1, is a typical example of the baseline network that uses 4 × 4 switches in the implementation. The network inputs and outputs are connected to an array of processor nodes. The nodes are renamed using a radix-2^n bit reversal scheme so that each of them is labeled with an m-bit binary address

$$a_{n-1} \cdots a_0 a_{2n-1} \cdots a_n \cdots a_{m-1} \cdots a_{m-n}$$

where $a_{m-1}a_{m-2} \cdots a_1\, a_0$ is the binary representation of its physical position. For instance, in a baseline network of $m = 4$ and $n = 2$, the processor nodes with consecutively physical positions are labeled 0000, 0100, 1000, 1100, 0001, 0101, etc. as shown in Fig. 1.

The m-bit binary number consists of m/n digits. Each digit is composed of n binary bits. For example, consider $m = 6$ and $n = 2$. There are three digits in the 6-bit binary number, and each digit consists of two binary bits. In the baseline network, establishment of a path $S \rightarrow D$ from the source (S) to the destination (D) can be accomplished through a *self-routing scheme.* We simply use the binary address of D as a *routing tag* to direct the connection. Let $S = s_{m-1}s_{m-2} \cdots s_1 s_0$ and $D = d_{m-1}d_{m-2} \cdots d_1 d_0$. Along the path to be established, every n binary bits (or every digit) $d_{ni+(n-1)}d_{ni+(n-2)} \cdots d_{ni+1}d_{ni}$ determine the setting of a switching element at stage i, where $0 \leq i \leq m/n$. Starting at the first stage, we set the first switch to which S is connected to the $(d_{n-1}d_{n-2} \cdots d_1 d_0)$th output port. We continue in this manner, switching on the $(d_{ni+(n-1)}d_{ni+(n-2)} \cdots d_{ni+1}d_{ni})$th output port at each stage i until the destination is reached.

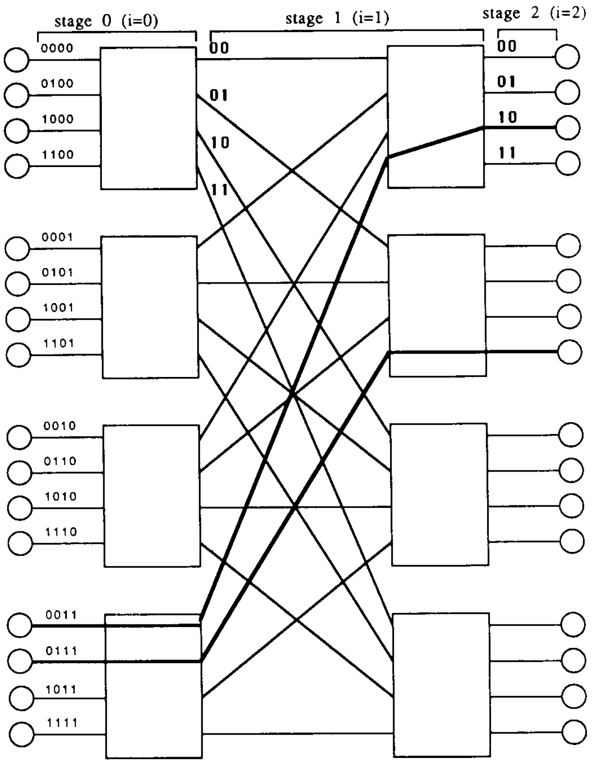


Fig. 1. A baseline network of $m = 4$, $n = 2$ and its routing scheme.

B. Detection of Connection Conflicts

The routing scheme, described in the previous subsection, allows us to rapidly establish individual connections by examining their destination addresses. However, simultaneous connections through the MIN may result in conflicts.

Definition: A connection conflict is a situation where two paths with different sources and destinations attempt to use the same communication links at certain stages at the same time.

A connection conflict implies potential message collisions at a certain communication link while two messages are simultaneously transmitted along the two paths. On the other hand, two paths are said to be conflict free if communication links traversed by the two paths are completely disjoint. It is worthwhile investigating the necessary and sufficient condition for ensuring conflict-free mappings of a collection of paths. First, let us define a function φ.

Definition: Let X and U be two m-bit binary numbers. $\varphi(X, U)$ is equal to $n \times$ (the maximum number of identical low-order digits of X and U).

For example, if we consider $m = 6$, $n = 2$, and $X = 00\ \underline{01}\ \underline{10}$, $U = 10\ \underline{01}\ \underline{10}$, then the maximum number of identical low-order digits equals 2. We have $\varphi(X, U) = n \times 2 = 4$.

Theorem 1: In the baseline network of size 2^m, two paths $X \rightarrow Y$ and $U \rightarrow V$, where $X \neq U$ and $Y \neq V$, are conflict free if and only if

$$\varphi(X, U)+\varphi(Y,V)<m.$$

Proof: The theorem is in fact a variation of a theorem in [20]. Its proof is omitted here. □

Notice that a necessary condition for conflict free is that the two paths must be established from two different sources to two different destinations. If two paths share a source node or a destination node, it is not considered a connection conflict. Theorem 1 indicates that the conflict-free property is determined by the network dimension m and is independent of the switch dimension n. In other words, the property is applicable to the baseline networks with different sizes of switches. The theorem also provides an efficient way of detecting connection conflicts; it allows us to check whether or not two paths have conflicts simply by examining the binary representations of the source and destination nodes. For example, consider a network of $m = 4$ and $n = 2$ as shown in Fig. 1. Suppose $X = 0011$, $Y = 10000$, $U = 0111$, and $V = 1101$. The two paths $X \rightarrow Y$ and $U \rightarrow V$ are conflict free, since $\varphi(X, U) + \varphi(Y, V) = 2 < m$.

III. The Conflict-Free Routing Scheme

The purpose of this section is to present a conflict-free routing scheme for constructing the quadtree communication structure on the baseline network. The quadtree structure provides an efficient means of accomplishing fast data search and distribution. We begin with introducing two basic functions which are needed for constructing the quadtree structure.

A. Two Functions and their Properties

Two bit-manipulation functions are introduced in this subsection. Throughout our discussion, unless specified otherwise, by a number we mean an m-bit binary number. The first function is a *bit-reverse* permutation, which is formally defined as follows.

Definition: For a number $X = x_{m-1}x_{m-2} \cdots x_1x_0$, ρ is a *bit-reverse* function which maps X in such a way that

$$\rho(X)=x_0x_1 \cdots x_{m-2}x_{m-1}.$$

For example, consider $m = 8$ and $X = 10110100$. We have $\rho(X) = \underline{00101101}$. The underline indicates those bits reversed by the function. ρ offers a valuable means of achieving conflict-free connections. This is formally described by the following theorem.

Theorem 2: Suppose that U and V are two different binary numbers. Then $U \rightarrow \rho(U)$ and $V \rightarrow \rho(V)$ are conflict free in the baseline network.

Proof: Let $U = u_{m-1}u_{m-2} \cdots u_0$ and $V = v_{m-1}\, v_{m-2} \cdots v_0$. Since $U \neq V$, we have $\varphi(U, V) = k$ for $0 \leq k \leq m - 1$. Thus, $\varphi(\rho(U),\rho(V)) < m - k$. As a result, we have

$$\varphi(U, V)+\varphi(\rho(U), \rho(V))<m.$$

Consequently, $U \rightarrow \rho(U)$ and $V \rightarrow \rho(V)$ are conflict free. □

For example, consider $m = 6$, $n = 2$, $U = 01\ \underline{10}\ \underline{11}$, and $V = 11\ \underline{10}\ \underline{11}$. We have $\rho(U) = 110110$ and $\rho(V) = 110111$. The two paths $U \rightarrow \rho(U)$ and $V \rightarrow \rho(V)$ are conflict free, since $\varphi(U, V) + \varphi(\rho(U), \rho(V)) = 4 < m$.

The second function is used to modify a binary number with a *double ring-tail* pattern described as follows.

Definition: For a number $X = x_{m-1}x_{m-2} \cdots x_0$, τ_h^q, where $0 \leq h < g \leq m - 1$, is a mapping that

$$\tau_h^g(X)=\overbrace{1 \cdots 1}^{m-g-1}\ 0\ \overbrace{x_{g-1} \cdots x_{h+1}}^{g-h-1}\ 0\ \overbrace{1 \cdots 1}^{h}.$$

In the special case that m is an odd number and $h = g = (m-1)/2$, the above expression becomes

$$\tau_h^g(X)=\overbrace{1 \cdots 1}^{h}\ 0\ \overbrace{1 \cdots 1}^{h}.$$

For example, consider $m = 8$ and $X = 01\underline{101}100$. We have $\tau_2^6\,(X) = 10\underline{101}011$. Note that in the above example, underlined bits remain unchanged. In case $m = 7$ and $h = g = 3$, we have $\tau_3^3\,(X) = 1110111$. To facilitate the presentation, we assume m is a positive even number in the following discussion. Next, we will show that the *double ring-tail* function, in combination with the *bit-reverse* function, can help us identify a class of conflict-free paths in the baseline network. This is illustrated by the following two theorems. For ease of presentation, we assume $\tau^m_{-1}(U) = U$.

Theorem 3: Suppose U and V are two different binary numbers. $\tau_h^{m-h-1}(\rho(U)) \neq \tau_h^{m-h-1}(\rho(V))$, where $0 \leq h \leq (m/2) - 1$. Then $\tau_{h-1}^{m-h}(U) \rightarrow \tau_h^{m-h-1}(\rho(U))$ and $\tau_{h-1}^{m-h}(V) \rightarrow \tau_h^{m-h-1}\,(\rho(V))$ are conflict free.

Proof: Provided $\tau_h^{m-h-1}(\rho(U)) \neq \tau_h^{m-h-1}(\rho(V))$, they must disagree at least in one bit position k, where $h + 1 \leq k \leq m - h - 2$. Thus, $\varphi(\tau_h^{m-h-1}\,(\rho(U)), \tau_h^{m-h-1}\,(\rho(V))) \leq k$. By Theorem 2,

$$\begin{aligned}\varphi(U, V)&\leq\varphi(\tau_h^{m-h-1}(U), \tau_h^{m-h-1}(V))\\ &=\varphi(\rho(\tau_h^{m-h-1}(\rho(U))), \rho(\tau_h^{m-h-1}(\rho(V))))\\ &<m-k.\end{aligned}$$

Consequently,

$$\varphi(U, V)+\varphi(\tau_h^{m-h-1}(\rho(U)), \tau_h^{m-h-1}(\rho(V)))<m. \quad \square$$

For example, consider $m = 6$, $n = 2$, $U = 11\ \underline{11}\ \underline{01}$, and $V = 10\ \underline{11}\ \underline{01}$. In case that $h = 0$, we have $\tau^6_{-1}(U) = U$, $\tau^6_{-1}(V) = V$, $\tau_0^5(\rho(V)) = 001110$, and $\tau_0^5(\rho(V)) = 001100$. The two paths $\tau^6_{-1}(U) \rightarrow \tau_0^5\,(\rho(U))$ and $\tau^6_{-1}(V) \rightarrow \tau_0^5\,(\rho(V))$ are conflict free, since $\varphi(U, V) + \varphi(\tau_0^5(\rho(U)), \tau_0^5(\rho(V))) = 4 < m$.

Theorem 4: Suppose U and V are two different numbers, $\tau_{h-1}^{m-h}(U) \neq \tau_{g-1}^{m-g}(V)$ and $\tau_h^{m-h-1}(\rho(U)) \neq \tau_g^{m-g-1}(\rho(V))$. Then $\tau_{h-1}^{m-h}(U) \rightarrow \tau_h^{m-h-1}\,(\rho(U))$ and $\tau_{g-1}^{m-g}(V) \rightarrow \tau_g^{m-g-1}(\rho(V))$, where $0 \leq h < g \leq (m/2) - 1$, are conflict free.

Proof: We need to consider two cases.

1) $h = 0$.

By the definition of τ, we have that the 0th bits of $\tau_0^{m-1}\,(\rho(U))$ and $\tau_g^{m-g-1}(\rho(V))$ are 0 and 1, respectively.

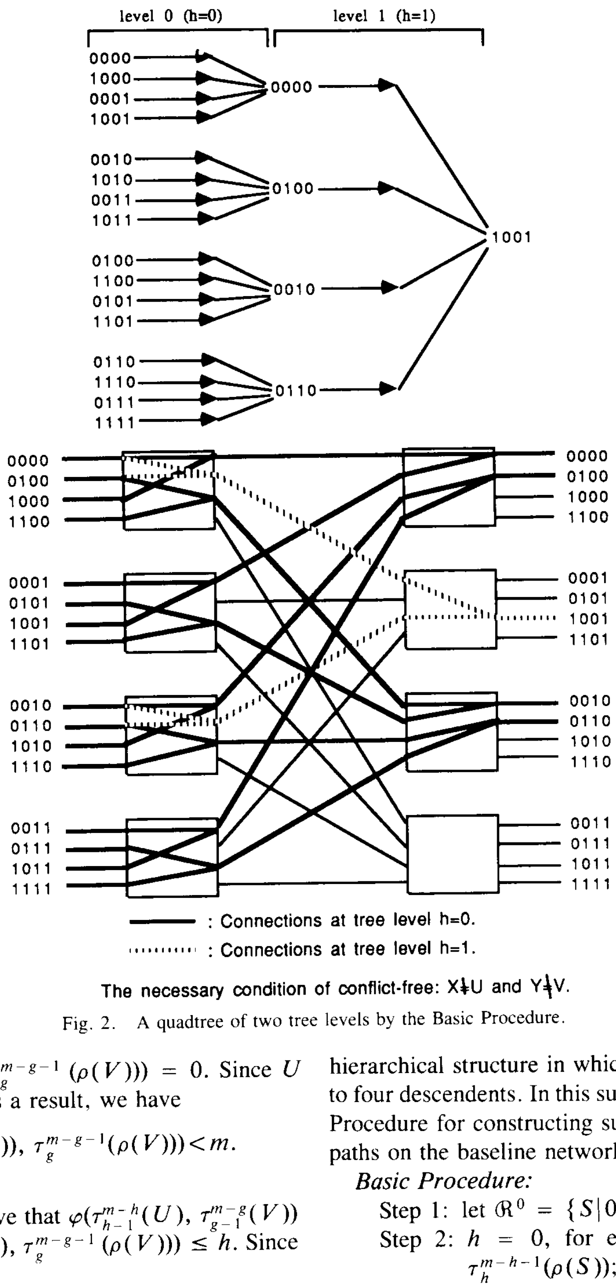


Fig. 2. A quadtree of two tree levels by the Basic Procedure.

Hence, $\varphi(\tau_0^{m-1}(\rho(U)), \tau_g^{m-g-1}(\rho(V))) = 0$. Since $U \neq V$, $\varphi(U, V) < m$. As a result, we have

$$\varphi(U, V)+\varphi(\tau_0^{m-1}(\rho(U)), \tau_g^{m-g-1}(\rho(V)))<m.$$

2) $h \neq 0$.

Because $h < g$, we must have that $\varphi(\tau_{h-1}^{m-h}(U), \tau_{g-1}^{m-g}(V)) \leq h - 1$ and $\varphi(\tau_h^{m-h-1}(\rho(U)), \tau_g^{m-g-1}(\rho(V))) \leq h$. Since $h < (m/2) - 1$,

$$\varphi(\tau_{h-1}^{m-h}(U), \tau_{g-1}^{m-g}(V)) + \varphi(\tau_h^{m-h-1}(\rho(U)), \tau_g^{m-g-1}(\rho(V)))<m. \qquad \square$$

For example, consider $m = 6$, $n = 2$, $U = 11\ \underline{11}\ \underline{00}$, and $V = 01\ \underline{11}\ \underline{00}$. In case that $g = 1$ and $h = 0$, we have $\tau^6_{-1}(U) = U$, $\tau_0^5(V) = V$, $\tau_0^5(\rho(U)) = 001110$, and $\tau_1^4(\rho(V)) = 101101$. The two paths $\tau^6_{-1}(U) \rightarrow \tau_0^5(\rho(U))$ and $\tau_0^5(V) \rightarrow \tau_1^4(\rho(V))$ are conflict free, since $\varphi(\tau^6_{-1}(U), \tau_0^5(V)) + \varphi(\tau_0^5(\rho(U)), \tau_1^4(\rho(V))) = 4 < m$.

B. Quadtree Graphs

The proposed routing scheme relies on the construction of a communication structure, called quadtree. A quadtree is a hierarchical structure in which each parent node is connected to four descendents. In this subsection, first we present a Basic Procedure for constructing such a structure with conflict-free paths on the baseline network.

Basic Procedure:

Step 1: let $\mathcal{R}^0 = \{S|0 \leq S \leq 2^m - 1\}$;
Step 2: $h = 0$, for every $S \in \mathcal{R}^0$, establish $S \rightarrow \tau_h^{m-h-1}(\rho(S))$;
Step 3: let $\mathcal{R}^{h+1} = \{\tau_h^{m-h-1}(\rho(S))|S \in \mathcal{R}^h\}$;
Step 4: $h = h + 1$, for every $S \in \mathcal{R}^h$, establish $S \rightarrow \tau_h^{m-h-1}(\rho(S))$;
Step 5: if $h < (m/2) - 1$, then repeat steps 3 and 4. $\square$

Fig. 2 gives an example of a quadtree achieved by using the procedure on the baseline network with 4 × 4 switches ($m = 4$ and $n = 2$). Note that the Basic Procedure can be applied to the baseline networks with different sizes of switches. Fig. 3 shows the mapping of the same quadtree structure on a baseline network with 2 × 2 switches ($m = 4$ and $n = 1$). Directed edges in Fig. 2 indicate source nodes and destination nodes of the paths to be established in the network. As can be seen, a quadtree constructed in this manner contains $m/2$ tree levels; the hth level, $0 \leq h \leq (m/2) - 1$, contains $4(m/2)$

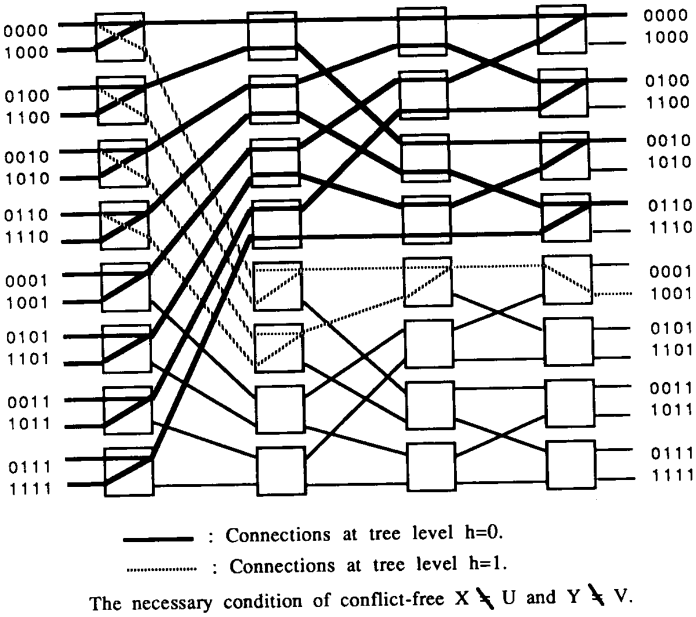


Fig. 3. Mapping of the quadtree structure on a baseline network with 2 × 2 switches.

$- h$ nodes. In addition, it roots at the processor node $R = 2^m - 2^{(m/2)} - 2^{(m/2)-1} - 1$. In fact, a quadtree can root at an arbitrary node through a proper transformation. This will be shown in the next subsection. Before that we show that a quadtree achieved by the Basic Procedure incurs no connection conflict in the network. We also show that if we reverse all the connections of a quadtree structure, the resulting quadtree incurs no conflict in the network.

Theorem 5: Suppose that $X \rightarrow Y$ and $U \rightarrow V$, $X \neq U$ and $Y \neq V$, are two connections of a quadtree through the Basic Procedure. Then $X \rightarrow Y$ and $U \rightarrow V$ are conflict free. Furthermore, $Y \rightarrow X$ and $V \rightarrow U$ are conflict free.

Proof: We need to consider two cases.

1) $X \rightarrow Y$ and $U \rightarrow V$ are located at the same level of the quadtree. That $X \rightarrow Y$ and $U \rightarrow V$ are conflict free has been shown in Theorem 3.

2) $X \rightarrow Y$ and $U \rightarrow V$ are located at different levels of the quadtree. That $X \rightarrow Y$ and $U \rightarrow V$ are conflict-free has been shown in Theorem 4.

Now that $X \rightarrow Y$ and $U \rightarrow V$ are conflict free, by Theorem 1, we have $\varphi(X, U) + \varphi(Y, V) < m$. $Y \rightarrow X$ and $V \rightarrow U$ are also conflict-free, since $\varphi(Y, V) + \varphi(X, U) < m$. □

In summary, the Basic Procedure essentially defines the hierarchy of the quadtree structure. It has been shown that nodes of a quadtree structure can be connected in two different directions, either parent-to-descendent or descendent-to-parent. Both result in conflict-free mapping on the baseline network.

C. Procedures for Ascending and Descending Messages

In this subsection, we present two formal procedures for ascending and descending messages through the quadtree structure. The two procedures are adapted from the Basic Procedure for accomplishing conflict-free communication through the network. The first procedure—Ascend Procedure—is used to select (or combine) data from all processor nodes and to store the result in the root node. Operations of data selection are initiated at the terminal nodes. Intermediate results are transferred toward the root through conflict-free paths. The second procedure—Descend Procedure—is used to transmit data, from the root node to all other processor nodes. Operations are initiated at the root node. Then data messages are propagated from high-level nodes to low-level nodes. We will show that a quadtree achieved by the two procedures can root at an arbitrary processor node R', where $0 \leq R' \leq 2^m - 1$. We assume that $Q = (2^m - 2^{(m/2)} - 2^{(m/2)-1} - 1) \oplus R'$.

Ascend Procedure:

```
Step 1: $\mathcal{R}^0 = \{S | 0 \leq S \leq 2^m - 1\}$;
Step 2: $h = 0$, for every $S \in \mathcal{R}^0$, send data to its parent
        node through $S \rightarrow \tau_h^{m-h-1}(\rho(S \oplus Q)) \oplus Q$;
Step 3: let $\mathcal{R}^{h+1} = \{\tau_h^{m-h-1}(\rho(S \oplus Q)) \oplus Q | S \in \mathcal{R}^h\}$;
Step 4: $h = h + 1$, for every $S \in \mathcal{R}^h$, fetch data
        transmitted from its four descendent nodes,
        perform operations on the data, and transmit the
        intermediate result to its parent node through $S$
        $\rightarrow \tau_h^{m-h-1}(\rho(S \oplus Q)) \oplus Q$;
Step 5: if $h < (m/2) - 1$, then repeat steps 3 and 4. □
```

An example of a quadtree structure by the Ascend Procedure is shown in Fig. 4. Note the correspondence of Exclusive OR between the two quadtrees of Figs. 2 and 4. The binary number in Fig. 4 can be obtained from that in Fig. 2 by applying the Exclusive OR operation with $Q = 0001$. We next introduce the dual procedure for disseminating data messages from the root node R' to all other processor nodes of the quadtree structure.

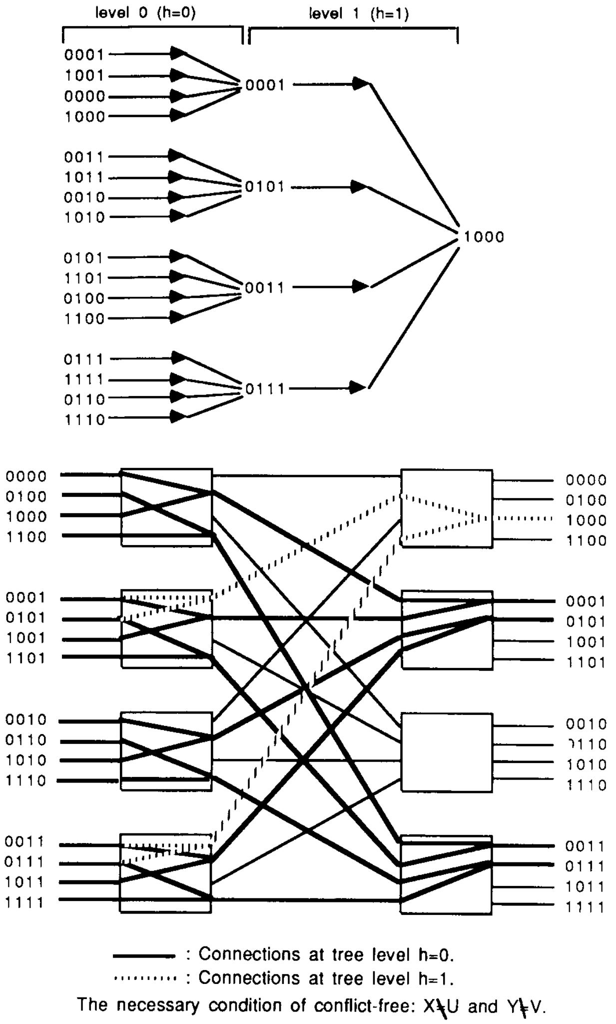


Fig. 4. A searching quadtree with $R' = 1000$, $Q = 0001$, and by the Ascend Procedure.

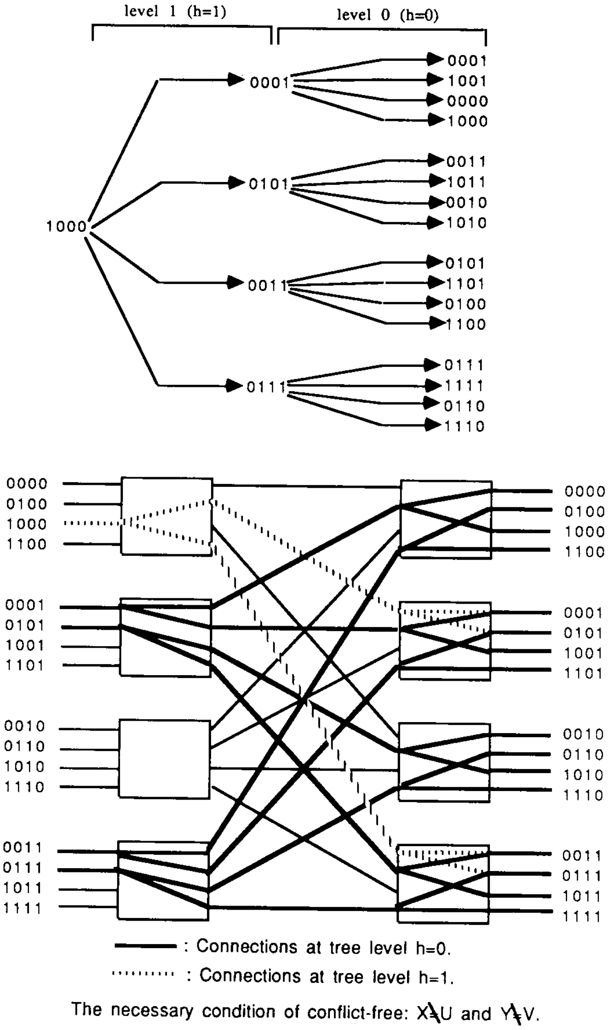


Fig. 5. A distributed quadtree with $R' = 1000$, $Q = 0001$, and by the Descend Procedure.

Descend Procedure:

```
Step 1: $h = (m/2) - 1$;
Step 2: $\mathcal{R}^h = \{S \mid 0 \le S \le 2^m - 1\}$;
Step 3: let $\mathcal{R}^{h-1} = \{\tau_h^{m-h-1}(\rho(S \oplus Q)) \oplus Q \mid S \in \mathcal{R}^h\}$;
Step 4: for every $S \in \mathcal{R}^{h-1}$, fetch data transmitted from
        its parent node (or prestored data in case of $h = (m/2) - 1$), modify the data
        if necessary, and pass them to its four descendent nodes through associated
        paths $S \rightarrow D$, where $D \in \{\tau_{h-1}^{m-h}(X) \mid X \in \mathcal{R}^h\}$
        and $\tau_h^{m-h-1}(\rho(D \oplus Q)) \oplus Q = S$;
Step 5: decrement $h$ by 1; if $h \ge 0$, then repeat steps 3 and 4.   □
```

It can be seen that some of the nodes may be traversed twice by a Descend Procedure; one occurs at a higher level $1 \le h \le (m/2) - 1$, and the other at the lowest level. For nodes of this type, the second data transmission becomes redundant if a Descend Procedure is used simply for broadcasting data. In case of this, the redundant transmission can be ignored. As an example, Fig. 5 shows the use of a quadtree structure for passing data from processor node 8 to all other nodes of a system with 16 nodes.

Now, let us show that quadtree structures established by the Ascend Procedure and the Descend Procedure result in no connection conflicts in the baseline network.

Theorem 6: Suppose that $X \rightarrow Y$ and $U \rightarrow V$, $X \ne U$ and $Y \ne V$, are two connections of a quadtree by the Ascend Procedure or the Descend Procedure. Then $X \rightarrow Y$ and $U \rightarrow V$ are conflict free.

Proof: Let $X' \rightarrow Y'$ and $U' \rightarrow V'$ be the two corresponding connections of $X \rightarrow Y$ and $U \rightarrow V$ of a quadtree by the Ascend Procedure; that is, $X = X' \oplus Q$, $Y = Y' \oplus Q$, $U = U' \oplus Q$, and $V = V' \oplus Q$. By Theorem 5, we have

$$\varphi(X', U') + \varphi(Y', V') < m.$$

Since $\varphi(X' \oplus Q, U' \oplus Q) = \varphi(X', U')$ and $\varphi(Y' \oplus Q,$

$V' \oplus Q) = \varphi(Y', V')$, we have

$$\varphi(X, U)+\varphi(Y, V)<m.$$

On the other hand, it is straightforward to show, by Theorem 5, that $X \rightarrow Y$ and $U \rightarrow V$ are conflict free in case that both are connections of a quadtree by the Descend Procedure. ☐

Our discussion up to this point is under the restriction that m has to be even. It requires certain modifications to make the procedures fit the networks of odd dimension. However, because of page limit, we will not describe these modifications here.

IV. Contraction and Stretch

The number of tree levels of a quadtree structure can be varied. This section describes methods for contracting and stretching the quadtree structure to suit needs of various parallel algorithm applications.

A. Contraction of the Quadtree Structure

The tree contraction is used to fuse adjacent levels of the quadtree structure. The resulting structure contains fewer tree levels, yet intermediate nodes are connected to more descendent nodes. We will show that the connections of a contracted tree structure still preserve the conflict-free property.

Suppose that ξ denotes the number of descendent nodes at each tree level (except the last one). In general, we assume ξ is a number of power of 2. Thus, we have

$$\text{the number of tree levels} = \left\lceil \frac{m}{d} \right\rceil$$

where $d = \log_2 \xi$ and $2 \le d \le m$ is a dimension for the function of *double ring-tail* τ. For example, $d = 2$ for the quadtree structure. Let h, $0 \le h \le \lceil m/d \rceil - 1$, be the number of tree levels. Then the number of nodes adjacent to the root $R = 2^m - 2^{(m/2)} - 2^{(m/2)-1} - 1$ at tree level $h = \lceil m/d \rceil - 1$ is equal to

$$\begin{cases} 2^d, & \text{if } m \bmod d = 0; \\ 2^{m \bmod d}, & \text{if } m \bmod d \ne 0. \end{cases}$$

The Ascend Procedure for the quadtree structure now can be modified into a General Procedure. It can adjust the number of tree levels by increasing the dimension d from 2 to m. Note that in the General Procedure for tree contraction, $p(h)$ and $q(h)$ are computed in the following manner:

1) if $0 \le h \le \lceil (m/d) \rceil - 2$, then $q(h) = \lfloor (h+1)d/2 \rfloor - 1$, $p(h) = m - (h+1)d + q(h) + 1$;
2) if $h = \lceil m/d \rceil - 1$, then $q(h) = (m/2) - 1$, $p(h) = m/2$.

General Procedure for Tree Contraction:

```
Step 1: let $\mathcal{R}^0 = \{S \mid 0 \le S \le 2^m - 1\}$;
Step 2: $h = 0$, for every $S \in \mathcal{R}^0$, establish $S \rightarrow \tau_{q(h)}^{p(h)}(\rho(S \oplus Q)) \oplus Q$;
Step 3: let $\mathcal{R}^{h+1} = \{\tau_{q(h)}^{p(h)}(\rho(S \oplus Q)) \oplus Q \mid S \in \mathcal{R}^h\}$;
Step 4: $h = h + 1$, for every $S \in \mathcal{R}^h$, establish $S \rightarrow \tau_{q(h)}^{p(h)}(\rho(S \oplus Q)) \oplus Q$;
Step 5: if $h < \lceil (m/d) \rceil - 1$, then repeat steps 3 and 4.  ☐
```

An example of the contracted tree structure of two levels is shown in Fig. 6 with $m = 6$ and $Q = 000000$. Note that contraction of a quadtree structure should incur no connection conflicts. This will be shown in Theorems 7 and 8.

B. Stretch of the Quadtree Structure

The counterpart of tree contraction is tree stretch. It is used to reduce the number of descendent nodes at the cost of increasing the number of tree levels. Here, we intend to stretch the quadtree structure by using the *double ring-tail* function τ. Without losing the conflict-free property, the first two tree levels have the same number of descendent nodes as the original quadtree structure. Let ξ represent the number of descendent nodes adjacent to each parent node and $d = 1$ in case of the stretched tree structure. Then,

$$\begin{cases} \xi = 2^{2d} = 4, & \text{for } 0 \le h \le 1; \\ \xi = 2^d = 2, & \text{for } 1 \le h \le m-3. \end{cases}$$

Hence, the number of tree levels equals $m - 2$. As the network dimension (m) increases, the number of levels of a stretched tree structure approaches to m. The Ascend Procedure can be modified by decreasing the dimension d from 2 to 1. Note that $p(h)$ and $q(h)$ are computed in the following manner:

1) if $h = 0$, then $q(h) = \lfloor (h+1)/2 \rfloor$, $p(h) = m - h + q(h) - 1$;
2) if $1 \le h \le m - 3$, then $q(h) = \lfloor (h+1)/2 \rfloor$, $p(h) = m - h + q(h) - 2$.

General Procedure for Tree Stretch:

```
Step 1: let $\mathcal{R}^0 = \{S \mid 0 \le S \le 2^m - 1\}$;
Step 2: $h = 0$, for every $S \in \mathcal{R}^0$, establish $S \rightarrow \tau_{q(h)}^{p(h)}(\rho(S \oplus Q)) \oplus Q$;
Step 3: let $\mathcal{R}^{h+1} = \{\tau_{q(h)}^{p(h)}(\rho(S \oplus Q)) \oplus Q \mid S \in \mathcal{R}^h\}$;
Step 4: $h = h + 1$, for every $S \in \mathcal{R}^h$, establish $S \rightarrow \tau_{q(h)}^{p(h)}(\rho(S \oplus Q)) \oplus Q$;
Step 5: if $h < m - 3$, then repeat steps 3 and 4.  ☐
```

An example of the stretched tree structure of four levels is shown in Fig. 7 with $m = 6$ and $Q = 000000$. We now prove the conflict-free property of the contracted and the stretched tree structures in Theorems 7 and 8, respectively.

Theorem 7: Suppose that U and V are two different numbers located at the same level h. $\tau_{q(h)}^{p(h)}(\rho(U)) \ne \tau_{q(h)}^{p(h)}(\rho(V))$, where 1) $0 \le h \le m - 3$, if $d = 1$; or 2) $0 \le h \le \lceil (m/d) \rceil - 1$, if $2 \le d \le m$. Then $U \rightarrow \tau_{q(h)}^{p(h)}(\rho(U))$ and $V \rightarrow \tau_{q(h)}^{p(h)}(\rho(V))$ are conflict free.

Proof: Since $\tau_{q(h)}^{p(h)}(\rho(U)) \ne \tau_{q(h)}^{p(h)}(\rho(V))$, they must disagree in the kth bit position, where $q(h) + 1 \le k \le p(h) - 1$. Thus, $\varphi(\tau_{q(h)}^{p(h)}(\rho(U)), \tau_{q(h)}^{p(h)}(\rho(V))) \le k$. By Theorem 2,

$$\begin{aligned} \varphi(U, V) &\le \varphi(\tau_{q(h)}^{p(h)}(U), \tau_{q(h)}^{p(h)}(V)) \\ &= \varphi(\rho(\tau_{q(h)}^{p(h)}(\rho(U))), \rho(\tau_{q(h)}^{p(h)}(\rho(V)))) \\ &< m - k. \end{aligned}$$

Consequently,

$$\varphi(U, V) + \varphi(\tau_{q(h)}^{p(h)}(\rho(U)), \tau_{q(h)}^{p(h)}(\rho(V))) < m. \qquad \square$$

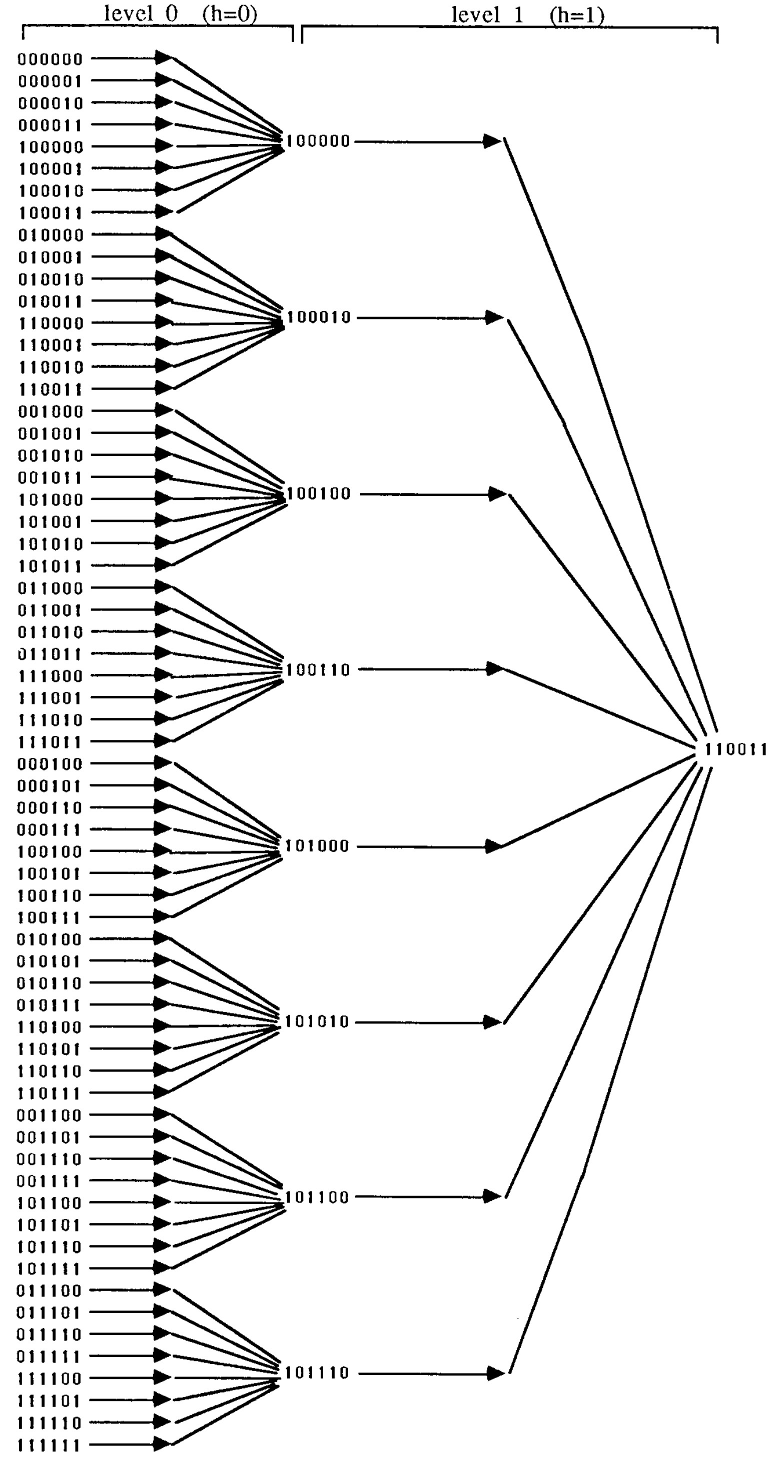


Fig. 6. A contracted tree structure of two tree levels.

Theorem 8: Suppose that U and V are two different numbers located at the tree level h and q, respectively. $\tau^{p(h)}_{q(h)}(\rho(U)) \neq \tau^{p(g)}_{q(g)}(\rho(V))$, where 1) $0 \leq h < g \leq m - 3$, if $d = 1$; or 2) $0 \leq h < g \leq \lceil (m/d) - 1 \rceil$, if $2 \leq d \leq m$. Then $U \rightarrow \tau^{p(h)}_{q(h)}(\rho(U))$ and $V \rightarrow \tau^{p(g)}_{q(g)}(\rho(V))$ are conflict free.

Proof: We consider two cases.

1) $h = 0$. By the definition of τ, we have that the 0th bits of $\tau^{p(0)}_{q(0)}(\rho(U))$ and $\tau^{p(g)}_{q(g)}(\rho(V))$ are 0 and 1, respectively. Hence, $\varphi(\tau^{p(0)}_{q(0)}(\rho(U)), \tau^{p(g)}_{q(g)}(\rho(V))) = 0$. Since $U \neq V$, $\varphi(U, V) < m$. Therefore, we have

$$\varphi(U, V) + \varphi(\tau^{p(0)}_{q(0)}(\rho(U)), \tau^{p(g)}_{q(g)}(\rho(V))) < m.$$

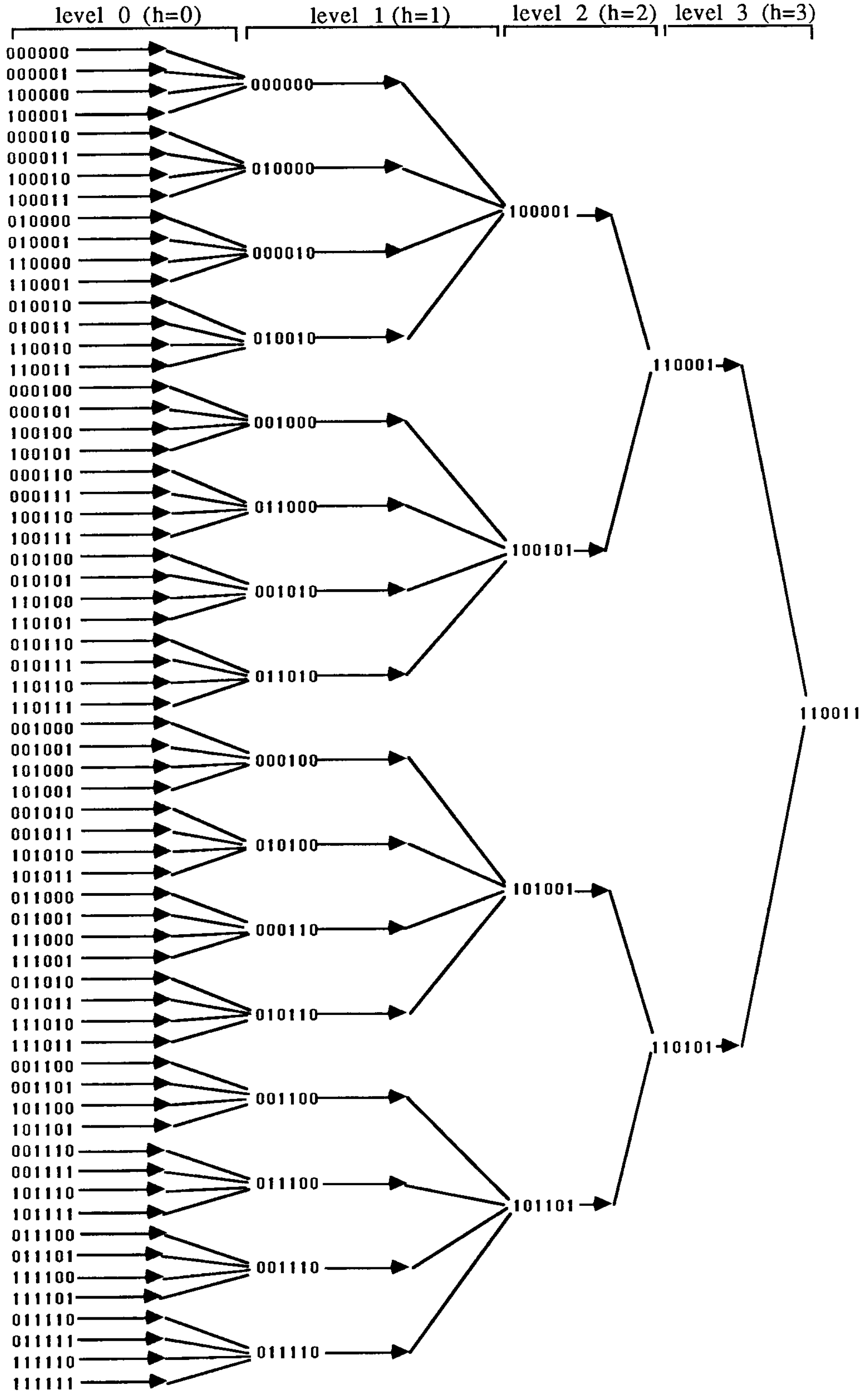


Fig. 7. A stretched tree structure of four tree levels.

2) $h \neq 0$. In case that $2 \le d \le m$, we must have $\varphi(U, V) \le q(h) - 1$ and $\varphi(\tau^{p(h)}_{q(h)}(\rho(U)), \tau^{p(g)}_{q(g)}(\rho(V))) \le q(h)$. Since $q(h) < \lceil m/d \rceil - 1$,

$$\varphi(U, V) + \varphi(\tau^{p(h)}_{q(h)}(\rho(U)), \tau^{p(g)}_{q(g)}(\rho(V))) \le 2q(h) - 1 < m.$$

In case that $d = 1$, if h is odd, then $p(h) + q(h) = m - 1$. Since $\varphi(U, V) \le q(h) - 1$ and $\varphi(\tau^{p(h)}_{q(h)}(\rho(U)), \tau^{p(g)}_{q(g)}(\rho(V))) \le p(h)$, we have

$$\varphi(U, V) + \varphi(\tau^{p(h)}_{q(h)}(\rho(U)), \tau^{p(g)}_{q(g)}(\rho(V))) \le p(h) + q(h) - 1 < m.$$

If h is even, then $p(h) + q(h) = m - 2$. Since $\varphi(U, V)$

$\leq p(h) + 1$ and $\varphi(\tau^{p(h)}_{q(h)}(\rho(U)), \tau^{p(g)}_{q(g)}(\rho(V))) \leq q(h)$, we have

$$\varphi(U, V) + \varphi(\tau^{p(h)}_{q(h)}(\rho(U)), \tau^{p(g)}_{q(g)}(\rho(V))) \leq p(h) + q(h) + 1 < m.$$

□

Theorems 7 and 8 show that the quadtree can be contracted and stretched at one's disposal, while its connections remain conflict free. Since the conflict-free property is independent of the switch dimension (n), the contracted and stretched tree structures can be mapped onto the baseline networks with different sizes of switching elements.

V. Analysis of the Quadtree Structure

The two procedures presented in the preceding sections allow us to adapt the quadtree structure in different ways. It is of particular interest and importance to determine the optimal adaptation for reducing the response time of combining and distributing data messages. We attempt to find out the optimal number of descendent nodes (ξ) of a quadtree structure such that the overall algorithm execution time can be minimized.

Our analysis is based on the following assumptions. We ignore the waiting time for resources. The execution time of each processor node is the sum of its computation time and communication time. We assume that a processor node is engaged in only one interprocessor communication at a time. A direct memory access, which requires only one-path communication cost, is used for processor nodes to access the remote memories. Provided no connection conflicts occur, the time required for one remote memory access is $(m/n) \times \mu$, where n is the switch dimension, m is the network dimension, and μ is a constant, We also assume that the time required for a local memory access is so small that it can be neglected. Let T_{para} be the execution time of parallel algorithms. Based on Section IV and the assumption that $2 \leq d \leq m$, we have

$$\text{the number of tree levels} = \left\lceil \frac{m}{d} \right\rceil = \left\lceil \frac{\log_2 2^m}{\log_2 \xi} \right\rceil = \lceil \log_\xi 2^m \rceil.$$

Let δ denote the number of remote memory accesses issued by each processor. The grain size (G) is defined as the number of arithmetic operations performed by each processor between two remote memory accesses. The time needed for each arithmetic computation unit is assumed to be t. Thus, we have

$$T_{\text{para}} = \left(Gt + \frac{\xi m \mu}{n}\right) \lceil \log_\xi 2^m \rceil + \overbrace{\left(Gt + \frac{\xi m \mu}{n}\right) + \cdots + \left(Gt + \frac{\xi m \mu}{n}\right)}^{\delta - 1}.$$

TABLE I

THE OPTIMAL NUMBER OF DESCENDENT NODES

$\mathcal{E}$	$\pi = 0$	$\pi = 1$	$\pi = 2$	$\pi = 3$	$\pi = 4$	$\pi = 5$
1024 (m=10)	2.42	5.05	9.60	17.08	28.95	47.25
256 (m=8)	2.37	4.76	8.66	14.76	24.00	37.61
64 (m=6)	2.30	4.37	7.53	12.16	18.77	27.97
16 (m=4)	2.18	3.84	6.13	9.21	13.28	16.00

In the above equation, the first term represents the execution time of the first message sent from descendent nodes to its parent node. The last $\delta - 1$ terms represent the execution time of the remaining $\delta - 1$ messages pipelined immediately after the first message is transmitted to the next tree level. In the above expression, we observe that the computation time T_{comp} and the communication time T_{comm} are

$$T_{\text{comp}} = (Gt)\lceil \log_\xi 2^m \rceil + (\delta - 1)(Gt),$$

and

$$T_{\text{comm}} = \left(\frac{\xi m \mu}{n}\right) \lceil \log_\xi 2^m \rceil + (\delta - 1)\left(\frac{\xi m \mu}{n}\right),$$

respectively. The ratio of T_{comp} and T_{comm} is defined as π. Hence,

$$\pi = \frac{T_{\text{comp}}}{T_{\text{comm}}} = \frac{nGt}{\xi m \mu}.$$

An optimal value ξ can minimize the total execution time so as to achieve a higher algorithm performance. To determine the optimal value ξ, we take the first order of differentiation of T_{para} with respect to ξ. By making the differentiation result equal 0, we have

$$(\ln \xi)^2 \frac{\xi m \mu}{n}(\delta - 1) + (\ln 2^m)\frac{\xi m \mu}{n}(\ln \xi - 1) - Gt(\ln 2^m) = 0.$$

Since $\pi = nGt/\xi m \mu$, the above equation can be reformulated into

$$(\ln \xi)^2(\delta - 1) + (\ln 2^m)(\ln \xi - 1) - \pi(\ln 2^m) = 0.$$

The optimal number of descendent nodes (ξ) can be evaluated by the Newton's method for different computation/communication ratios (π) and for different network dimensions (m). The results are shown in Table I and Fig. 8 for $\delta = 2$. In Fig. 8, we observe that the quadtree structure should be stretched for parallel algorithms with smaller computation/communication ratios. On the other hand, for parallel algorithms with larger computation/communication ratios, we need to contract the quadtree structure to minimize the overall algorithm execution time.

VI. Conclusions

We have presented a conflict-free routing scheme for efficient data search and distribution on a class of MIN-based parallel and distributed computing systems. The heart of our routing scheme is the quadtree communication structure. The quadtree structure suggests a general approach to mapping parallel algorithms that require intensive interprocessor com-

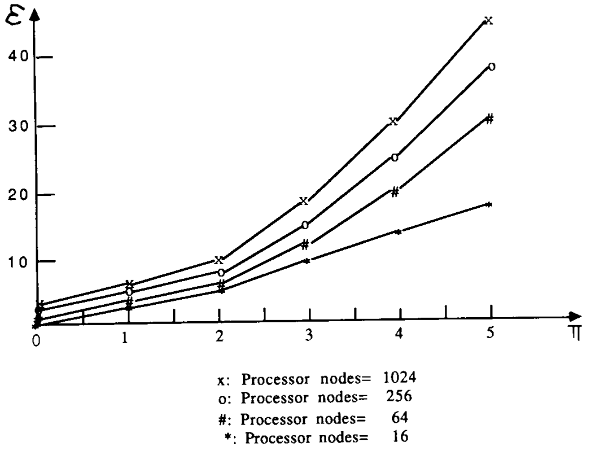


Fig. 8. The optimal number of descendent nodes versus the computation/communication ratios.

munication for selecting data from many different sources and for disseminating data from a single source. We have shown that while performing these two operations through the quadtree structure, the two procedures—Ascend Procedure and Descend Procedure—incur no communication link conflict. By properly merging messages and efficiently replicating data, the quadtree structure can accomplish required communications in $O(\log_4 M)$ parallel steps, where M is the network size.

We have shown that the size of a quadtree communication structure can be contracted and stretched by adjusting the number of descendent nodes. Contraction and stretch of the quadtree structure do not affect its conflict-free property. We have also presented the relationship between the computation/communication ratio of various parallel algorithms and the number of tree levels. Finally, we have examined their joint effect on the response time of combining and distributing data messages. This analysis helps us determine the optimal adaptation of the quadtree for minimizing the overall algorithm execution time.

REFERENCES

[1] W. Lin, M. Thazhuthaveetil, and C. R. Das, "A parallel matrix inversion algorithm with dynamic communication structures," in *Proc. 1988 Int. Conf. Supercomput.*, vol. 2, pp. 460–466.

[2] T. L. Sheu and W. Lin, "Mapping linear programming algorithms onto the Butterfly Parallel Processor," in *Proc. 1988 Int. Conf. Supercomput.*, vol. 2, pp. 452–459.

[3] C. L. Wu and T. Y. Feng, "On a class of multistage interconnection networks," *IEEE Trans. Comput.*, vol. C-29, pp. 694–702, Aug. 1980.

[4] T. Y. Feng, "Data manipulating functions in parallel processors and their implementations," *IEEE Trans. Comput.*, vol. C-23, pp. 309–318, Mar. 1974.

[5] D. H. Lawrie, "Access and alignment of data in an array processor," *IEEE Trans. Comput.*, vol. C-24, pp. 1145–1155, Dec. 1975.

[6] K. E. Batcher, "The flip network in STARAN," in *Proc. 1976 Int. Conf. Parallel Processing*, pp. 65–71.

[7] L. R. Goke and G. J. Lipovski, "Banyan networks for partitioning multiprocessor systems," in *Proc. 1st Symp. Comput. Architecture*, Dec. 1973, pp. 175–189.

[8] M. C. Pease III, "The indirect binary *n*-cube multiprocessor array," *IEEE Trans. Comput.*, vol. C-26, pp. 458–473, May 1977.

[9] S. Thanawastien and V. P. Nelson, "Interference analysis of shuffle/exchange networks," *IEEE Trans. Comput.*, vol. C-30, pp. 545–556, Aug. 1981.

[10] R. H. Thomas, "Behavior of the Butterfly Parallel Processor in the presence of memory hot spots," in *Proc. 1986 Int. Conf. Parallel Processing*, pp. 46–50.

[11] K. E. Batcher, "STARAN series E.," in *Proc. 1977 Int. Conf. Parallel Processing*, Aug. 1977, pp. 140–143.

[12] G. F. Pfister and V. A. Norton, "'Hot spot' contention and combining in multistage interconnection networks," *IEEE Trans. Comput.*, vol. C-34, pp. 943–948, Oct. 1985.

[13] A. Gottlieb, R. Grishman, C. Kruskal, K. McAuliffe, L. Rudolph, and M. Snir, "The NYU Ultracomputer—Designing an MIMD shared memory parallel computer," *IEEE Trans. Comput.*, vol. C-32, pp. 175–189, Feb. 1983.

[14] H. J. Siegel. L. J. Siegel, F. C. Kemmerer. P. T. Mueller, Jr., H. E. Smalley, Jr., and S. D. Smith, "PASM: A partitionable SIMD/MIMD system for image processing and pattern recognition," *IEEE Trans. Comput.*, vol. C-30, pp. 934–947, Dec. 1981.

[15] R. J. McMillen, G. B. Adams, and H. J. Siegel, "Performance and implementation of 4 × 4 switching nodes in an interconnection network for PASM," in *Proc. Int. Conf. Parallel Processing*, Aug. 1981, pp. 229–233.

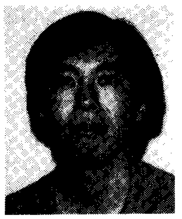
[16] S. Cheemalavagu and M. Malek, "Analysis and simulation of banyan interconnection networks with 2 × 2, 4 × 4, and 8 × 8 switching elements," in *Proc. Real-Time Systm. Symp.*, Dec. 1982, pp. 83–89.

[17] W. Crowther, J. Goodhue, E. Starr, R. Thomas, W. Milliken, and T. Blackadar, "Performance measurements on a 128-node Butterfly Parallel Processor," in *Proc. 1985 Int. Conf. Parallel Processing*, pp. 531–540.

[18] R. Rettberg, and R. Thomas, "Contention is no obstacle to shared-memory multiprocessing," *Commun. ACM*, vol. 29, pp. 1202–1212, Dec. 1986.

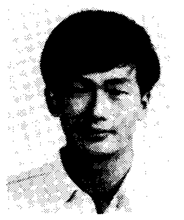
[19] T. J. LeBlanc, "Shared memory versus message passing in a tightly coupled multiprocessor: A case study," in *Proc. 1986 Int. Conf. Parallel Processing*, pp. 463–466.

[20] W. Lin and C. L. Wu, "Reconfiguration procedures for a polymorphic and partitionable multiprocessor," *IEEE Trans. Comput.*, vol. C-35, pp. 910–916, Oct. 1986.



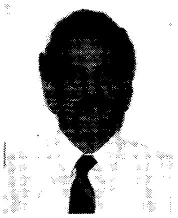
Woei Lin (S'80–M'85) received the B.S. degree in control engineering from National Chiao-Tung University, Taiwan, in 1978, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Texas at Austin, in 1982 and 1985, respectively

He is currently an Assistant Professor in the Department of Electrical and Computer Engineering, University of New Mexico, Albuquerque, Prior to joining the University of New Mexico, he was a faculty member at the Pennsylvania State University from 1985 to 1988. He had served in the army in Taiwan as an officer partly for digital system maintenance from 1978 to 1980. He was a Teaching Assistant from 1981 to 1984 and a Research Assistant from 1984 to 1985 at the University of Texas at Austin. His research interests include multiprocessor systems, parallel algorithms, interconnection networks, and parallel programming environments.



Tsang-Ling Sheu (S'84) received the B.S. degree in electrical engineering from National Cheng-Kung University, Taiwan, in 1981, the M.S. degree in electrical engineering from Virginia Polytechnic Institute and State University, Blacksburg, in 1985, and the Ph.D. degree in computer engineering from the Department of Electrical Engineering, The Pennsylvania State University, University Park, in 1989.

From, 1981 to 1983, he served as a Electronics Officer in the navy in Taiwan. Since January 1986, he has been working as a Research Assistant in the Computer Engineering program at The Pennsylvania State University. His research interests include parallel/distributed processing, computer architecture, fault-tolerant computing, parallel algorithms, data communication, and performance evaluation.



Chita R. Das (S'84–M'86) received the M.Sc. degree in electrical engineering from regional engineering College, Rourkela, Sambalpur University, India in 1981, and the Ph.D. degree in computer science from the Center for Advanced Computer Studies, University of Southwestern Louisiana, Lafayette, in 1986.

He is an Assistant Professor in the Computer Engineering Program at The Pennsylvania State University. His research interests include parallel/distributed computing, performance evaluation, and fault-tolerant computing.

Tse-yun Feng (S'61–M'67–SM'75–F'80) received the B.S. degree from the National Taiwan University, Taipei, Taiwan, the M.S. degree from Oklahoma State University, Stillwater, and the Ph.D. degree from the University of Michigan, Ann Arbor, all in electrical engineering.

He was on the Faculty of the Department of Electrical and Computer Engineering, Syracuse University, Syracuse, NY, Wayne State University, Detroit, MI, and the Department of Computer and Information Science, Ohio State University, Columbus. He is now Binder Professor of Computer Engineering at The Pennsylvania State University, University Park. He has extensive technical publications in the area of parallel and concurrent processors, interconnection networks, computer architecture, and switching theory.

Dr. Feng is editor-in-chief of a new journal, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED PROCESSING and was editor-in-chief of IEEE TRANSACTIONS ON COMPUTERS, 1982–1986. He has edited a number of Sagamore Computer Conference Proceedings and Interconnection Networks for Parallel and Distributed Processing. In addition, he has also edited special issues for the IEEE TRANSACTIONS ON COMPUTERS and *ACM Computing Surveys*. He has been an invited speaker to various organizations and served as a consultant or reviewer to several companies and publishers. He has received a number of awards and honorary recognitions for his technical contributions and scholarship. He is also active professionally. He was President of the Computer Society of the IEEE (1979–1980), Chairman of the International Conference on Computers and Applications, among others. He is currently Chairman of the Distinguished Visitors' Program of the Computer Society of the IEEE and Chairman of the International Conference on Parallel Processing.

Chuan-Lin Wu (S'74–M'78–SM'85) received the B.S.E.E. and M.S. degrees from National Chiao Tung University, Taiwan, and the Ph.D. degree from Wayne State University, Detroit, MI.

Since September 1981, he has been a faculty member with the University of Texas at Austin. His research interests include parallel processing, computer architecture, network management, expert systems, and computer-aided VLSI design. He has published more than 100 technical papers on these and related area. As a frequently invited speaker for seminars and short courses, he has also served as a consultant.