ELSEVIER

Journal of Systems Architecture 46 (2000) 919–930

JOURNAL OF SYSTEMS ARCHITECTURE

www.elsevier.com/locate/sysarc

Efficient path-based multicast in wormhole-routed mesh networks

Tzung-Shi Chen [a,*], Chih-Yung Chang [b], Jang-Ping Sheu [c]

[a] Department of Information Management, Chang Jung University, Tainan 711, Taiwan, ROC
[b] Department of Information Science, Aletheia University, Taipei, Taiwan, ROC
[c] Department of Computer Science and Information Engineering, National Central University, Chung-Li, Taiwan, ROC

Received 22 February 1999; received in revised form 7 September 1999; accepted 14 October 1999

Abstract

The capability of multidestination wormhole allows a message to be propagated along any valid path in a wormhole-routed network conforming to the underlying base routing scheme. The multicast on the path-based routing model is highly dependent on the spatial locality of destinations participating in multicasting. In this paper, we propose two proximity grouping schemes for efficient multicast in wormhole-routed mesh networks with multidestination capability by exploiting the spatial locality of the destination set. The first grouping scheme, *graph-based proximity grouping*, is proposed to group the destinations together with locality to construct several disjoint sub-meshes. This is achieved by modeling the proximity grouping problem to graph partitioning problem. The second one, *pattern-based proximity grouping*, is proposed by the pattern classification schemes to achieve the goal of the proximity grouping. By simulation results, we show the routing performance gains over the traditional Hamiltonian-path routing scheme.

Keywords: Interconnection networks; Mesh networks; Multicast; Wormhole routing; Parallel computing

1. Introduction

Multicast, an important communication mechanism, is frequently used in a wide variety of applications containing parallel algorithms, scientific parallel computing, and so forth, for message-passing multicomputers. There are a lot of works paying attention to designing multicasting algorithms on a variety of interconnection networks, such as hypercube and mesh networks. They aimed at designing collective communication [10] which is mainly based on the switching techniques containing store-and-forward, virtual cut-through, or wormhole routing methods [4,11].

In general, the multicasting problem, one-to-many routing, can be modeled by three routing schemes: tree-based, unicast-based, and path-based routing [10]. By using the tree-based routing, the multicast algorithms were proposed in [7,18,17]. The tree-based multicasting highly relies on finding a tree, rooted at the source node, connecting each destination node on the given target network. The source message is propagated to each destination along the constructed tree. In [3,9,19], they proposed the multicasting on mesh networks under the unicast-based routing. The

* Corresponding author. Fax: +886-6-2780657.
E-mail addresses: chents@mail.cju.edu.tw (T.-S. Chen), changcy@email.au.edu.tw (C.-Y. Chang), sheujp@csie.ncu.edu.tw (J.-P. Sheu).

1383-7621/00/$ - see front matter
PII: S1383-7621(99)00049-1

major disadvantage to the unicast-based routing is that there are a lot of extra startup latencies incurred while delivering the message from each intermediate node to another.

Recently, in the literature [2,5,8,12,13,15,16], they turned attention to multicasting the message under the path-based routing on wormhole-routed networks. They used the system architectures with multidestination capacity to try to improve the data transmission performance over that using the unicast-based routing schemes. The trip-based model [15] was proposed on networks with arbitrarily topologies. For fault-tolerance to networks, Tseng et al. [16] proposed the Euler-path-based multicasting model on general networks. Lin et al. [8] presented a Hamiltonian-path deadlock-free routing approach to efficiently performing multicast on mesh networks. Panda et al. [12] proposed a hierarchical leader-based scheme on mesh networks. Fan and King [5] proposed a turn grouping method to solve the multicasting problem on mesh networks under the turn model [6].

In this paper, we aim at the mesh networks with path-based wormhole routing model as the target architectures. For efficient multicasting, one of the major concerns is on exploiting the spatial locality of the destination nodes participating in the multicasting [12]. Two proximity grouping schemes proposed here are for efficient multicast in wormhole-routed mesh networks with multidestination capability. The first grouping scheme, *graph-based proximity grouping*, is proposed to group the destinations together with locality to construct several disjoint sub-meshes by modeling the proximity grouping problem to graph partitioning problem. The second one, *pattern-based proximity grouping*, is proposed by the pattern classification schemes to achieve the goal of the proximity grouping. By exploiting the spatial locality of the destination set, we can efficiently perform the multicast in wormhole-routed mesh networks conforming to any path-based wormhole routing. After performing proximity grouping, multicasting for all of partitioned groups can be individually performed in parallel. As a result, the total multicast latency can be improved. Here we show that our proposed techniques are superior to the simple Hamiltonian-path routing scheme by simulation results.

The rest of the organization of this paper is stated as follows. Section 2 introduces the system model of mesh multicomputers and states the motivations of this paper. In Section 3, we present the multicast algorithm with two proximity grouping schemes, the graph-based proximity grouping and the pattern-based proximity grouping. The graph-based proximity grouping is proposed to partition destination set based on a graph model so that the proximity destinations are constructed into a group within a sub-mesh. The pattern-based proximity grouping is next presented to explore the destination set in order to form groups with proximity relations based on the pattern classification schemes. Simulation results are discussed in Section 4. Conclusions are finally summarized in Section 5.

2. Preliminaries

In this section, we first introduce our system model for multidestination wormhole-routed mesh multicomputers. We then state the related work and the motivations for exploiting the spatial locality of the destination set for multicasting in mesh networks.

2.1. System model

The mesh multicomputer system is composed of nodes, each node is a computer with its own processor and local memory, and communication link, each directed link connects to two neighboring nodes through network [8]. The node architecture of a mesh multicomputer system is as shown in Fig. 1. A common component of nodes in a new generation multicomputer is a router. It can handle the message communication entering, leaving, and passing through the node. The architecture of the mesh network system provides the wormhole routing with multidestination capability.

We denote $M_{k_1 \times k_2 \times \cdots \times k_n}$ as an n-dimensional mesh where $(x_1, x_2, \ldots, x_n)$ is one of the nodes, $0 \leqslant x_i \leqslant k_i - 1$ for $1 \leqslant i \leqslant n$. Symbol k_i is denoted as the maximum number of the i-dimension in the n-dimensional mesh. Let $\mathrm{SM}(l_1 \ldots h_1, l_2 \ldots h_2, \ldots,$

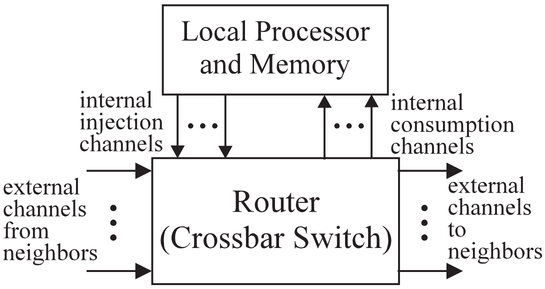


Fig. 1. A node architecture with multidestination routing.

$l_n \ldots h_n$) denote a sub-mesh in $M_{k_1 \times k_2 \times \cdots \times k_n}$, where $(x_1, x_2, \ldots, x_n)$ is one of the nodes $0 \leqslant l_i \leqslant x_i \leqslant h_i \leqslant k_i - 1$ for $1 \leqslant i \leqslant n$. In this paper, we use the routing model conforming to the path-based routing with multidestination capability.

2.2. Related work and motivations

We first state the related work in this section. Here, we only pay attention to the multicasting on the basis of path-based routing model in mesh networks. Lin et al. [8] presented a Hamiltonian-path deadlock-free routing approach to efficiently perform multicasting on mesh networks. Using the dual-path multicast routing, the source node delivers the message along two subnetworks, high-channel network and low-channel network, on a mesh. We illustrate their idea with an example as shown in Fig. 2. In this 8×8 mesh, there are a source node $S = (2, 5)$, denoted by a solid point, and 8 destination nodes, denoted by the empty

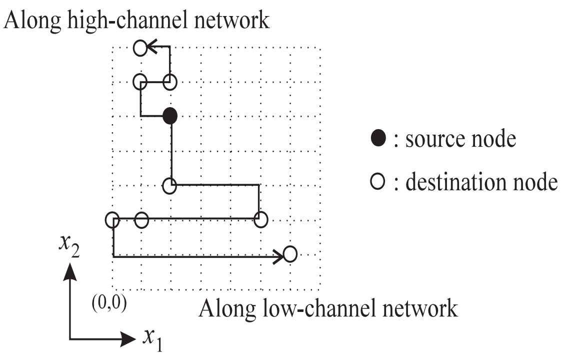


Fig. 2. Hamiltonian-path routing with dual-path.

points, as shown in Fig. 2. The source node (2, 5) sends the message to three destinations (1, 6), (2, 6), and (1, 7) along the high-channel network in turn. Meanwhile, the source node (2, 5) sends the message to five destinations (2, 3), (5, 2), (1, 2), (0, 2), and (6, 1) along the low-channel network in turn. Note that the routing path between two nodes is the shortest. It is easy to see that these two multidestination messages are passed and traversed with 23 communication links.

Panda et al. [12] proposed a hierarchical leader-based scheme. Their scheme is to partition the destination set into disjoint groups. A leader for each group is selected to form a leader set. These leaders of the leader set are partitioned repeatedly into disjoint leader groups. Thus there exists a hierarchy structure of groups. After that, the source node sends the message to their members within the same group. By recursively delivering the message, the multicasting can be completed.

Fan and King [5] proposed a turn grouping method based on the turn model [6]. By establishing a pturn-dependence graph [5], they use the graph coloring algorithm to group the destination nodes into several groups conforming to the turn model. Then a leader for each group is selected to generate a leader set. The source node sends the message to all the nodes in the leader set via a multidestination message. The leader node delivers the multidestination message to the members of its groups simultaneously while it received a message from the source.

For any path-based routing scheme, it is responsible for exploit the spatial locality of destination set on a mesh network. This allows multicasting to avoid incurring path thrashing on routing over proximity groups. That is, one of the nodes on each group with spatial locality is responsible for delivering the source message to the other nodes on its group so as to minimize the communication cost. Thus, minimizing the multicast latency on a mesh network is our main goal in this paper.

We consider the same example, as shown in Fig. 2, to illustrate our idea and motivation to this problem. Suppose that we partition the destination set into disjoint subsets, each surrounded by a sub-mesh network. These sub-mesh networks are

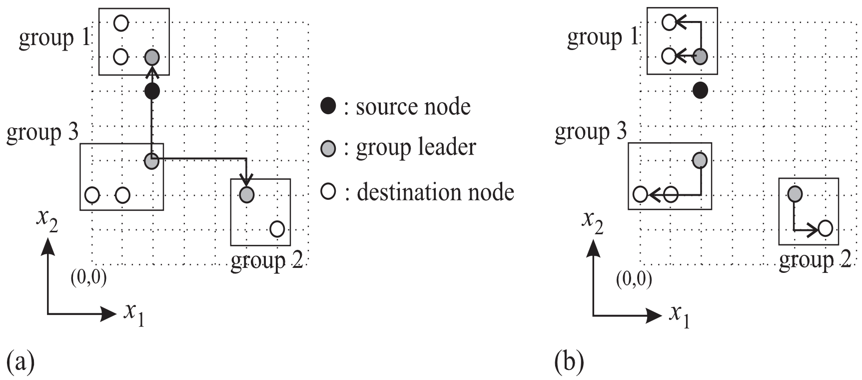


Fig. 3. Multicast with proximity grouping by using Hamiltonian dual-path routing. (a) Source-to-leader routing in step 1. (b) Leader-to-destination routing in step 2.

disjoint so that the destination nodes within each sub-mesh has proximity relationship. As depicted in Fig. 3(a), we can partition the destination set into three disjoint groups 1, 2, and 3 which are bounded by the respective sub-meshes. Assume that we use the dual-path Hamiltonian routing scheme in this system. We first select the leaders $(2,6)$, $(5,2)$, and $(2,3)$ from groups 1, 2, and 3, respectively. The source node first sends the multidestination message to these leaders according to the dual-path Hamiltonian routing scheme as shown in Fig. 3(a) . After each leader received the message, it individually sends the multidestination message in parallel to each destination within its corresponding sub-mesh as shown in Fig. 3(b). Thus, we not only use 15 communication links but also reduce the transmission time to this example, compared to the traditional Hamiltonian routing scheme as shown in Fig. 2.

3. Multicasting

In this section, we shall propose the multicast algorithm along with two proximity grouping schemes for exploiting the spatial locality of destinations participating in a multicasting on mesh networks.

We first formulate the multicast algorithm that a source node S wants to multicast a source message to k destinations on a wormhole-routed n-dimensional mesh network. We denote the set containing k destinations as $D = \{d_1, d_2, \ldots, d_k\}$. In the following, we describe the multicast algorithm with two-level routing.

Algorithm 1 (Multicast-Algorithm ($M_{k_1 \times k_2 \times \cdots \times k_n}$, S, D)).

Step 1. Given the destination set D on the mesh $M_{k_1 \times k_2 \times \cdots \times k_n}$, we first group them into several groups with proximity relations via the grouping schemes to be presented in the two subsequent sections. A group containing destination nodes is surrounded within a sub-mesh so that these sub-meshes generated by the groups are disjoint.

Step 2. One leader L_i for each group G_i is selected to generate a set L of group leaders. Here the leader L_i on group G_i is chosen by evaluating the distance between the source node S and each node on group G_i such that the distance is minimum.

Step 3. Then, we deliver a multidestination message from source to each group leader on the set L based on the system routing scheme, called the *source-to-leader routing*. The multidestination message routing conforms to one of the deadlock-free routing schemes, such as e-cube [4], turn-model [6], Hamiltonian path-based [8], base-routing-conformed-path (BRCP) model [12].

Step 4. In terms of conforming to the underlying routing as in Step 3, we perform multicast in parallel on each group with proximity relations, called the *leader-to-destination routing*, because these groups without intersection are disjoint. Finally, we complete the multicasting on the mesh network.

In the following two sections, we shall discuss two grouping schemes for exploiting the spatial locality of the destination set on the mesh networks. These schemes for multicasting are used in Step 1 of Algorithm 1.

3.1. Graph-based proximity grouping

In this section, we will propose the graph-based proximity grouping scheme. Some definitions and terms are first introduced.

Definition 1 (*Proximity graph*). For a set D of destination nodes in an n-dimensional mesh, a weighted *proximity graph* is defined as $G=(V,E)$. The set of vertices, V, consists of the destination nodes $D=\{d_1,d_2,\ldots,d_k\}$. The set of edges, E, consists of the edges of $e=(d_i,d_j)$ where its weight, the distance between two nodes, is defined as $W(e)=\sum_{1\leqslant p\leqslant n}|x_p-y_p|$, $d_i=(x_1,x_2,\ldots,x_n)$ and $d_j=(y_1,y_2,\ldots,y_n)$, for d_i, $d_j\in D$ and $d_i\neq d_j$.

We illustrate how the proximity can be constructed with an example below. Consider the same example as shown in Fig. 2, in the previous section. According to the definition of the proximity graph, the constructed proximity graph is shown in Fig. 4 where the weight of an edge is calculated and is placed on the corresponding edge.

We first illustrate our idea for exploiting the spatial locality of destinations participating in a multicast operation. The grouping is in a way that we partition the corresponding proximity graph into groups surrounded by disjoint sub-mesh networks such that the destination nodes on each group have spatial locality. Once the proximity graph is constructed, we partition the graph into several proximity groups. We first partition the whole mesh into two disjoint sub-meshes (groups) along a hyperplane such that the weight across the edges of the graph is maximum. Under the key consideration of minimizing communication cost in intra-group, we have to partition the group

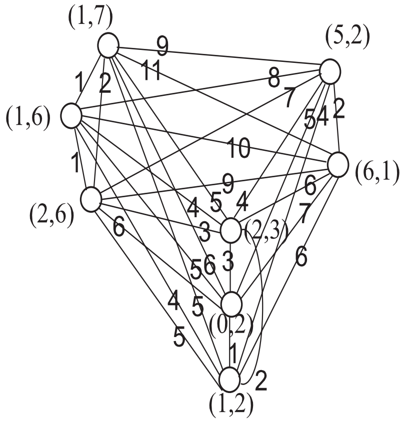


Fig. 4. An example of the proximity graph.

along the edges with the great total amount of communication overhead into two sub-groups. For each group, we recursively partition it into two disjoint sub-meshes until one specified constraint is satisfied. There is a corresponding sub-mesh generated for each group containing the destination set D'. The sub-mesh is defined as the form $\mathrm{SM}_{D'}(l_1\ldots h_1,l_2\ldots h_2,\ldots,l_n\ldots h_n)$ surrounding all the nodes in D', where l_i is the minimum value of i-dimensional on each ith element and h_i is the maximum value of i-dimensional on each ith element for all nodes. We define the ratio of the cardinality of set D' to the total number of nodes in $\mathrm{SM}_{D'}(l_1\ldots h_1,l_2\ldots h_2,\ldots,l_n\ldots h_n)$ as

$$R_{\mathrm{SM}_{D'}}=\frac{|D'|}{\sum_{1\leqslant i\leqslant n}|h_i-l_i|},$$

where we denote $|D'|$ as the cardinality of the set D'. Finally, we give a threshold value to terminate the partitioning process if $R_{\mathrm{SM}_{D'}}\geqslant T$, where T is denoted by a threshold value. For example, $T=1/2$, namely, the number of destination nodes are at least one half of the total number of nodes within the sub-mesh SM, i.e., $R_{\mathrm{SM}_{D'}}\geqslant T=1/2$. For instance, the destination set $D_1=\{(1,6),(1,7),(2,6)\}$ in group 1 is surrounded by $\mathrm{SM}_1(1\ldots 2,6\ldots 7)$ in Fig. 3(a). The ratio of the cardinality of set D_1 to the total number of nodes in $SM_1(1\ldots 2,6\ldots 7)$ is 3/4.

We formalize the above descriptions as the following algorithm.

Algorithm 2 (Graph-based proximity grouping (sub-mesh $\mathrm{SM}_D(l_1 \ldots h_1, l_2 \ldots h_2, \ldots, l_n \ldots h_n)$, destination set)).

Step 1. If the ratio R_{SM_D} is greater than or equal to the threshold value T, this algorithm is terminated; otherwise, go to Step 2.

Step 2. For each hyperplane $x_i = H_i$, $l_i \leqslant H_i \leqslant h_i$, $1 \leqslant i \leqslant n$, we partition the corresponding proximity graph G with D along $x_i = H_i$ in this sub-mesh. As a result, there are two sets D_1 and D_2 generated via the partitioning of the hyperplane $x_i = H_i$ such that $d = (y_1, y_2, \ldots, y_n) \in D_1$ with $y_i \geqslant H_i$ and $d' = (z_1, z_2, \ldots, z_n) \in D_2$ with $z_i < H_i$, where $D_1 \cup D_2 = D$ and $D_1 \cap D_2 = \phi$.

Step 3. For each partitioning, we evaluate their cost (weight) across the proximity graph G.

Step 4. Determine the partitioning hyperplane $x_i = H_{\max}$ so that there exists the maximum weight in the crossed edges when partitioning the SM into two disjoint sub-meshes SM_{D_1} and SM_{D_2} with destination sets D_1 and D_2, respectively.

Step 5. Recursively perform the algorithm on the sub-meshes SM_{D_1} and SM_{D_2} along with D_1 and D_2, respectively.

At the beginning of Algorithm 2, the first call is:

Graph-based proximity grouping $(M_{k_1 \times k_2 \times \cdots \times k_n}, D)$,

where $M_{k_1 \times k_2 \times \cdots \times k_n}$ is the target machine and D is the destination set for multicasting. We discuss how to derive the time complexity of Algorithm 2 below. We have $|D|$ destination nodes. Thus, it takes the time complexity $O(|D|^2 n)$ to construct a proximity graph G, a complete graph with $|D|$ vertices and $|D|(|D|-1)/2$ edges, where n is the dimension of one node in the mesh. The most time-consuming part of this algorithm is in Step 2 with a couple of recursive calls. In Step 2, we use a hyperplane to partition the graph in order to generate two disjoint sets D_1 and D_2. That is, we have to evaluate its cost of edges across the proximity graph G. Thus, it needs to take the worst time complexity $O(|D|^2)$ for all edges. The first call needs to take the time complexity $O(k_1|D|^2 n + k_2|D|^2 n + \cdots + k_n|D|^2 n)$. Each level of recursive call needs to take the same time as the first recursive call. For the worst case, we have $|D|$ groups, each group only containing one destination. Therefore, the total time complexity of this algorithm is

$$O((k_1|D|^2 n + k_2|D|^2 n + \cdots + k_n|D|^2 n)|D|) = O((k_1 + k_2 + \cdots + k_n)|D|^3 n).$$

Obviously, this algorithm has polynomial time complexity.

We reconsider the example described in Section 2. Here we assume the threshold value $T = 1/2$. The first call to this example is expressed by

Graph-based proximity grouping $(M_{8\times 8}, D)$,

where $D = \{(1,6), (2,6), (1,7), (2,3), (5,2), (1,2), (0,2), (6,1)\}$. Within the first call, the constructed proximity graph is shown in Fig. 4. We can use the hyperplane $x_2 = 4$ to partition the whole mesh into two sub-mesh networks surrounding the destination nodes as shown in Fig. 5(a). Here $D_1 = \{(1,6), (2,6), (1,7)\}$ and $D_2 = \{(2,3), (5,2), (1,2), (0,2), (6,1)\}$. As partitioning across the edges depicted in Fig. 5(b), its total weight (communication cost) equal to 93 is maximum. Thus, we have two sub-meshes $SM_{D_1}(1 \ldots 2, 6 \ldots 7)$ and $SM_{D_2}(0 \ldots 6, 1 \ldots 3)$. Then we recursively call the Algorithm 2. It is terminated after the call

Graph-based proximity grouping

$(\mathrm{SM}_{D_1}(1 \ldots 2, 6 \ldots 7), D_1)$,

because $R_{\mathrm{SM}_{D_1}} = 3/4 \geqslant 1/2$. The other call

Graph-based proximity grouping

$(\mathrm{SM}_{D_2}(0 \ldots 6, 1 \ldots 3), D_2)$,

is proceeding to partition the sub-mesh $\mathrm{SM}_{D_2}(0 \ldots 6, 1 \ldots 3)$ into two sub-meshes, $\mathrm{SM}_{D_3}(0 \ldots 2, 2 \ldots 3)$ and $\mathrm{SM}_{D_4}(5 \ldots 6, 1 \ldots 2)$, along hyperplane $x_1 = 4$ as depicted in Fig. 6(a). The corresponding proximity graph is partitioned across the edges with the weight 32 as depicted in Fig. 6(b). Two recursive calls are performed and then are terminated. Therefore, we have three groups 1, 2, and 3 to the destination set D surrounded by their respective sub-meshes $\mathrm{SM}_{D_1}(1 \ldots 2, 6 \ldots 7)$, $\mathrm{SM}_{D_3}(0 \ldots 2, 2 \ldots 3)$ and $\mathrm{SM}_{D_4}(5 \ldots 6, 1 \ldots 2)$. Finally, we use Algorithm 1

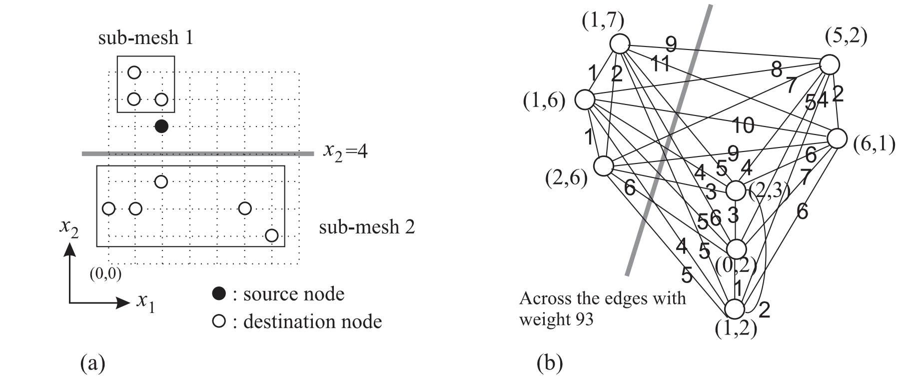


Fig. 5. Partitioning the proximity graph of D along hyperplane $x_2 = 4$. (a) The source node and destination set. (b) The corresponding proximity graph.

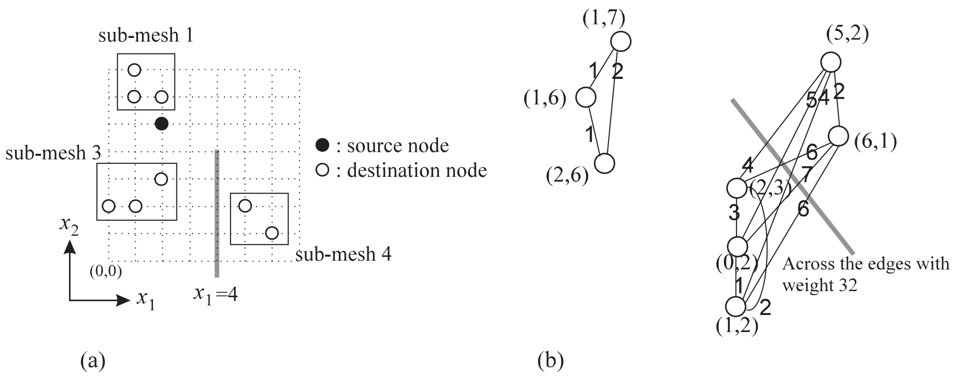


Fig. 6. Partitioning the proximity graph of D_2 along hyperplane $x_1 = 4$. (a) The source node and destination set. (b) The corresponding proximity graph.

to perform the multicasting conforming to the dual-path Hamiltonian routing on the mesh network as illustrated in Section 2.

3.2. Pattern-based proximity grouping

In this section, we will propose the pattern-based proximity grouping scheme. We use the pattern classification approaches [14] to grouping the destination nodes so that these groups have the proximity property. The proximity among destination nodes is determined by the distance functions. Here we focus the classification problem on the *maximin* (maximum–minimum)-distance approach proposed in [1,14].

We introduce the basic concepts of *maximin*-distance approach for classifying patterns in an n-dimensional space. The *maximin*-distance approach is a heuristic procedure based on the Euclidean distance concept. Here the Euclidean distance stands for the value of

$$\mathrm{Dist}(x,y) = \sqrt{(x_1-y_1)^2+(x_2-y_2)^2+\cdots+(x_n-y_n)^2},$$

where two nodes $x=(x_1,x_2,\ldots,x_n)$ and $y=(y_1,y_2,\ldots,y_n)$ are in an n-dimensional mesh. Below in this section, the distance means the Euclidean distance for convenience. Assume that we will classify the nodes in destination set D into several irregular clusters, not a sub-mesh form. Arbitrarily let a node c_1 be a cluster representative. Next, we determine the farthest node, called a cluster representative c_2. In the third step, we compute the distance from each remaining node $d \in D$ to c_1 and c_2. We save the summation of the minimum Euclidean distances $\mathrm{Dist}(d, c_1)$ and $\mathrm{Dist}(d, c_2)$. Then, we select the maximum value from these minimum distances. If this distance with maximum value is an appreciable fraction of the distance, $\mathrm{Dist}(c_1, c_2)$, between c_1 and c_2 (here assuming at least one half of the average of the summation distance for each pair on a set of the cluster representatives), the cluster representative is c_3. Otherwise, this algorithm is terminated. In the next step, the above procedure is repeated until the new maximum distance at a particular step fails to satisfy the condition for the creation of a new cluster representative.

Therefore, we have a set C of cluster representatives by the above procedure. Assume that the representative of group G_i is c_i for $1 \leqslant i \leqslant j$ if we have j groups. The remaining nodes $d \in D$ and $d \notin C$ is added to the group G_i if we have the minimum distance $\mathrm{Dist}(d, c_i)$, i.e., $\min_{c_i \in C} \mathrm{Dist}(d, c_i)$. Thus, all of proximity groups are generated based on the above maximin-distance approach. We know that it is possible to generate irregular groups, not sub-mesh form via this method.

In the following, we will discuss how to use the maximin-distance approach to generate the proximity groups surrounding by disjoint sub-mesh networks.

Algorithm 3 (Pattern-based proximity grouping ($M_{k_1 \times k_2 \times \cdots \times k_n}$, destination set D)).

Step 1. We apply the above-mentioned procedure maximin-distance approach [1,14] to generate the set C of cluster representatives.

Step 2. We require to add the remaining destination nodes to the proximity groups according to their dimension ordering, $x_1, x_2, \ldots, x_n$, in sequence. While we assign a remaining node p to its corresponding sub-mesh, we have to examine whether joining node p to a sub-mesh to form a new sub-mesh can incur the intersection among some sub-meshes. If it has occurred, let the node p be a new cluster representative so that the set C is set to $C \cup \{p\}$.

Step 3. We continuously repeat the operation of Step 2 until all of the remaining nodes in D are scanned. Finally, we have several disjoint sub-meshes surrounding some destination nodes.

After grouping, we are able to use the multicast algorithm presented previously to perform multicasting with two-level routing. If these sub-meshes are not disjoint, they may incur contention or deadlock as performing multicast. Hence we have to avoid contention or deadlock when the data transmission is performed independently on each sub-mesh. In general, the number of generated groups on Algorithm 3 is slightly larger than the number of clusters generated by the traditional maximin-distance approach.

In what follows, we discuss how to derive the time complexity of Algorithm 3. We have $|D|$ destination nodes. Within Step 1, we use the maximin-distance approach for generating the set of C of cluster representatives. Initially, we have one representative c_1. Then, we have to find c_2 out. It needs the time $\mathrm{O}(n)$ to compute the Euclidean distance $\mathrm{Dist}(x, y)$ between two nodes x and y. Thus, it costs the time complexity $\mathrm{O}(n(|D|-1) + (|D|-1))$ to find out c_2. To be continued, clearly, we need the time $\mathrm{O}(n(|D|-i) \times i + (|D|-i))$ to find the representative c_i out. The worst case is that all of the nodes in D are the cluster representatives. Thus, it needs the time complexity shown in Step 1.

$$\mathrm{O}((n(|D|-1) + (|D|-1)) + (n(|D|-2) \times 2 + (|D|-2)) + \cdots + (n \times 1 \times (|D|-1) + 1)) = \mathrm{O}(n|D|^3).$$

Then, we add the remaining destination nodes into their corresponding groups G_i with their corresponding cluster representatives c_i. The worst case of this operation is needed with the time complexity $\mathrm{O}(n|D|)$. Followed by Steps 2 and 3, we need to visit each destination node at once. Hence, the worst case of these two steps needs the time complexity $\mathrm{O}(n|D|)$. Therefore, this algorithm takes the time complexity

$$O(n|D|^3) + O(n|D|) + O(n|D|) = O(n|D|^3).$$

Clearly, this approach has polynomial time complexity.

Consider an example as shown in Fig. 7, where there are 18 destination nodes in a destination set D. We arbitrarily select a node $c_1 = (0, 0)$ as the first cluster representative. Next, we determine the farthest node $(7, 2)$, called a cluster representative c_2, to node c_1 with the maximum Euclidean distance

$$\text{Dist}(c_2, c_1) = \sqrt{(7-0)^2 + (2-0)^2} = \sqrt{53}.$$

In the third step, we compute the distance from each remaining node $d \in D$ to c_1 and c_2. We save the summation of the minimum distances $\text{Dist}(d, c_1)$ and $\text{Dist}(d, c_2)$. We have the maximum minimum distance

$$\begin{aligned} L_{\text{dist}} &= \text{Dist}(c_3, c_1) + \text{Dist}(c_3, c_2) \\ &= \sqrt{(3-0)^2 + (5-0)^2} + \sqrt{(3-7)^2 + (5-2)^2} \\ &= 5 + \sqrt{34} \end{aligned}$$

for the node $c_3 = (3, 5)$. Due to $L_{\text{dist}} \geqslant Dist(c_2, c_1)/2 = \sqrt{53}/2$, the distance L_{dist} is larger than one half of the average of distance of c_1 and c_2. Thus, the node c_3 is named a cluster representative. Next, we compute the distance from each remaining node $d \in D$ to c_1, c_2, and c_3. Thus, we save the summation of the minimum distances $\text{Dist}(d, c_1)$, $\text{Dist}(d, c_2)$, and $\text{Dist}(d, c_3)$. We know that the distance is not larger than one half of the average of the summation distance of c_1 and c_2, c_1 and c_3, and c_2 and c_3. Next, we simply assign each remaining node to its nearest cluster representative by computing the minimum distance for each pair $d \in D$ and c_i, $1 \leqslant i \leqslant 3$. Thus, this procedure is terminated. Namely, we have three clusters 1, 2, and 3 with the cluster representatives c_1, c_2, and c_3, respectively, as shown in Fig. 7.

Next, the three clusters 1, 2, and 3 generated are surrounded by the three sub-meshes 1, 2, and 5, respectively, as shown in Fig. 8. We can see that the two sub-meshes 1 and 5 are intersection, not disjoint. Under the consideration of disjoint grouping, we add the destination nodes to each proximity group according to the dimension ordering, first x_1, then x_2. That is, the destination node ordering is of lexicographical order. While processing, the node $(3, 2)$ is to be appended into the group 3, incurring the situation of the sub-meshes 1 and 3 being not disjoint. Thus, a new cluster representative $c_4 = (3, 2)$ is generated. After repeating the same procedure, the new group generated is surrounded by the new sub-mesh 4. The final result with four sub-meshes 1, 2, 3, and 4 is illustrated in Fig. 8. After grouping, we use Algorithm 1 to perform the multicast with two-level routing: source-to-leader routing as shown in Fig. 9(a) and leader-to-destination routing as shown in Fig. 9(b) by using Hamiltonian dual-path routing.

4. Simulation results

In this section, we will discuss the simulation results. We compare our developed schemes with the Hamiltonian dual-path routing while we adopt the Hamiltonian dual-path scheme to route the

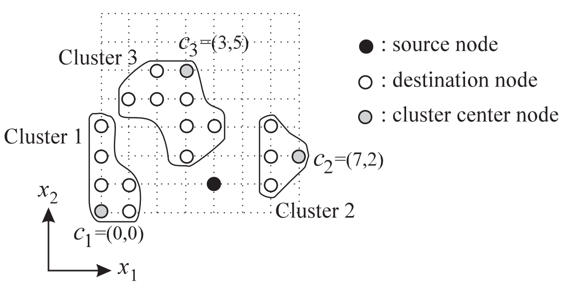


Fig. 7. An example of the pattern classification.

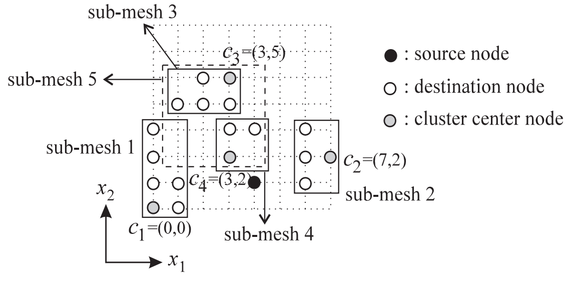


Fig. 8. The result of the pattern-based proximity grouping.

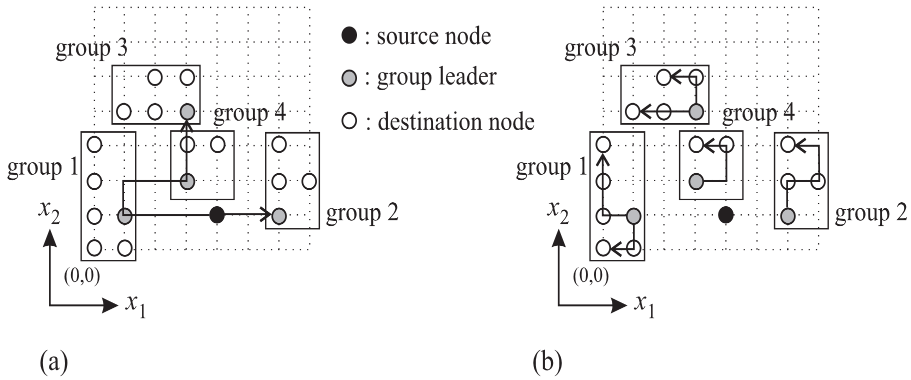


Fig. 9. Multicasting via pattern-based proximity grouping by using Hamiltonian dual-path routing. (a) Source-to-leader routing. (b) Leader-to-destination routing.

message on both source-to-leader and leader-to-destination steps in our algorithms.

Here we first give some assumptions to the parameters of system architecture and our algorithm in our simulation. The message startup latency t_s is 1.0 μs . One link propagation delay t_l is 5.0 ns and the router delay t_r on each node is 20.0 ns for each routing flit. For all of the multicasting on our simulation, the message size is assumed to be 100 flits. Here assume the threshold value $T = 4/5$ used in the graph-based proximity grouping scheme.

The simulation results are shown in Figs. 10 and 11. We simulate our proposed multicast algorithms with graph-based proximity grouping (Graph-Based) and pattern-based proximity grouping (Pattern-Based), compared to the Hamiltonian dual-path scheme (Ham-Path) on 16 × 16 mesh. In Fig. 10, the destinations are randomly generated. We show that the multicast latency by our proposed schemes is improved over the traditional Hamiltonian dual-path scheme. In Fig. 11, we discuss the cases as the destination nodes are constructed with proximity relationship. Due to the destination nodes with spatial locality, a great amount of the multicast latency in Fig. 11 can be reduced as compared with the results depicted in Fig. 10.

There exist different number of groups generated when applying the graph-based and pattern-based proximity grouping schemes we proposed.

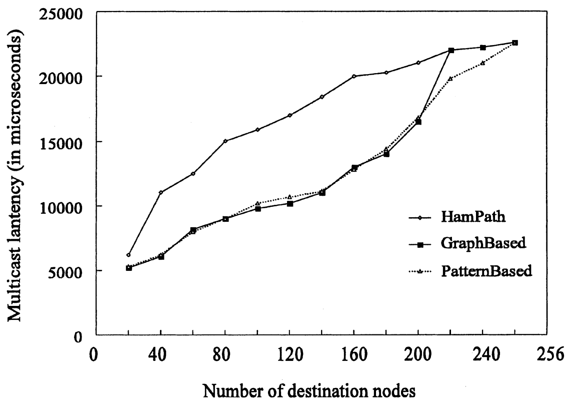


Fig. 11. Comparison of Hamiltonian-path, graph-based and pattern-based proximity grouping schemes to destinations with proximity relations on 16 × 16 mesh.

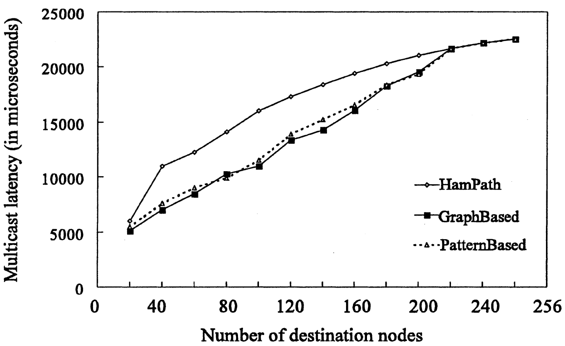


Fig. 10. Comparison of Hamiltonian-path, graph-based and pattern-based proximity grouping schemes to random destinations on 16 × 16 mesh.

The pattern-based grouping scheme has nice property related to spatial locality. It may, however, produce a lot of groups because it introduces the irregular shape grouping. We have to partition the irregular shape into disjoint sub-meshes. However, the grouping using the graph-based proximity grouping is highly dependent on the selection of the threshold value T, namely, the partitioning criterion. Hence the number of proximity groups generated by the two proximity grouping schemes maybe different. As shown in Figs. 10 and in 11, the number of proximity groups and their located positions affect the multicast latency. When the number of destination nodes is increasing to the number of all nodes, we assume that the whole mesh is a group. Namely, we use the Hamiltonian dual-path scheme to perform multicasting. It turns out that, via simulation study, the multicast performance of our proposed multicast algorithms based on exploiting the spatial locality of destination set is improved over that of the previous one [8].

5. Conclusions

In this paper, we proposed two proximity grouping schemes for exploiting the spatial locality to the destination set on multicasting for mesh networks. The first grouping scheme, *graph-based proximity grouping*, was proposed to partition the destination set to form disjoint sub-meshes with proximity nodes. The second one, *pattern-based proximity grouping*, was proposed by the pattern classification schemes to achieve the goal of the proximity grouping. Using these two proposed grouping schemes, we presented the efficient multicast algorithm with two major steps, source-to-leader routing and leader-to-destination routing. There are two advantages for exploiting spatial locality of the destination set on mesh networks. One is that the number of traversed links can be reduced to improve the communication performance. The other is that the message propagation on these disjoint sub-mesh networks can be performed in parallel independently so as to reduce the communication time. Our proposed grouping schemes can be used in any path-based multicast routing as a preprocessing scheme so as to minimize multicast latency. Finally, the simulation results are given to show that applying proximity grouping schemes to multicasting for mesh networks can improve the performance over the traditional Hamiltonian-path routing scheme.

Acknowledgements

This work was supported in part by the National Science Council of Republic of China under Grant NSC-87-2213-E-309-004. In addition, the authors would like to thank the anonymous referees for their critical review and valuable comments on correcting several errors and greatly improving the overall presentation of this paper.

References

[1] B.G. Batchelor, B.R. Wilkins, Method for location of clusters of patterns to initialise a learning machine, Electronics Lett. 5 (20) (1969) 481–483.

[2] T.-S. Chen, N.-C. Wang, C.-P. Chu, Path-based multicast communication in wormhole-routed star graph multicomputers, in: Proceedings of the 1998 International Conference on Parallel and Distributed Systems, 1998, pp. 350–357.

[3] L.D. Coster, N. Dewulf, C.-T. Ho, Efficient multi-packet multicast algorithms on meshes with wormhole and dimension-ordered routing, in: Proceedings of International Conference on Parallel Processing, 1995, pp. 137–141.

[4] W.J. Dally, C.L. Seitz, Deadlock-free message routing in multiprocessor interconnection networks, IEEE Trans. Comput. C-36 (5) (1987) 547–553.

[5] K.-P. Fan, C.-T. King, Turn grouping for efficient multicast in wormhole mesh networks, in: Proceedings of Sixth Symposium on Frontiers of Massively Parallel Computing, 1996, pp. 50–57.

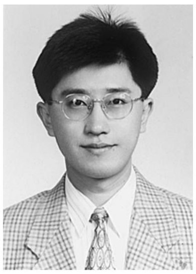
[6] C.J. Glass, L.M. Ni, The turn model for adaptive routing, in: Proceedings of International Symposium on Computer Architecture, 1992, pp. 278–287.

[7] Y. Lan, A.H. Esfahanian, L.M. Ni, Multicast in hypercube multiprocessors, J. Parallel and Distributed Comput. 8 (1990) 30–41.

[8] X. Lin, P.K. Mckinley, L.M. Ni, Deadlock-free multicast wormhole routing in 2D mesh multicomputers, IEEE Trans. Parallel and Distributed Syst. 5 (8) (1994) 793–804.

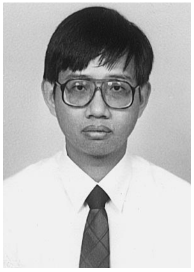
[9] P.K. Mckinley, H. Xu, A.H. Esfahanian, L.M. Ni, Unicast-based multicast communication in wormhole-routed networks, IEEE Trans. Parallel and Distributed Syst. 5 (12) (1994) 1252–1265.

[10] P.K. Mckinley, Y.-J. Tsai, D.F. Robinson, Collective communication in wormhole-routed massively parallel computers, IEEE Comput. 28 (12) (1995) 39–50.
[11] L.M. Ni, P.K. McKinley, A survey of wormhole routing techniques in direct networks, IEEE Comput. 26 (2) (1993) 62–76.
[12] D.K. Panda, S. Singal, P. Prabhakaran, Multidestination message passing mechanism conforming to base wormhole routing scheme, in: Proceedings PCRCW'94, Lecture notes in Computer Science, No. 853, Springer, Berlin, 1994, pp. 131–145.
[13] D.F. Robinson, P.K. Mckinley, B.H.C. Cheng, Path-based multicast communication in wormhole-routed unidirectional torus networks, J. Parallel and Distributed Comput. 45 (2) (1997) 104–121.
[14] J.T. Tou, R.C. Gonzalez, Pattern Recognition Principles, Addison-Wesley, Reading, MA, 1974.
[15] Y.-C. Tseng, D.K. Panda, T.-H. Lai, A trip-based multicasting model in wormhole-routed networks with virtual channels, IEEE Trans. Parallel and Distributed Syst. 7 (2) (1996) 138–150.
[16] Y.-C. Tseng, T.-Y. Juang, M.-H. Yang, An Euler-path-based multicasting model for wormhole-routed networks: Its applications to damaged 2D tori and meshes, in: Proceedings of International Performance, Computing, and Communication Conference, 1997, pp. 444–450.
[17] Y.-C. Tseng, T.-Y. Juang, M.-C. Du, Some heuristics and experiments for building a multicasting tree in a high-speed network, in: Proceedings of HPC Asia '97 Conference, 1997, pp. 248–253.
[18] J.-P. Sheu, M.-Y. Su, A multicast algorithm for hypercube multiprocessors, in: Proceedings of International Conference on Parallel Processing III, 1992, pp. 18–22.
[19] S.-Y. Wang, Y.-C. Tseng, C.-W. Ho, Efficient multicast in wormhole-routed 2D mesh/torus multicomputers: A network-partitioning approach, in: Proceedings of Symposium of Frontiers of Massively Parallel Computation, 1996, pp. 42–29.



Tzung-Shi Chen was born in Taiwan, Republic of China, on 16 December, 1966. He received the B.S. degree in computer science and information engineering from Tamkang University, Taiwan, in June 1989 and Ph.D. in computer science and information engineering from the National Central University, Taiwan, in June 1994. Since June 1996, he joined the faculty of the Department of Information Management, Chung Jung University, Tainan, Taiwan, Republic of China, as an Associate Professor. His current research interests include interconnection networks, parallelizing compilers, and distributed systems. Dr. Chen is a member of the IEEE Computer Society.

Chih-Yung Chang was born in Tainan, Taiwan, on 18 March, 1964. He received the B.S. degree in information engineering from the Chung-Yung University, Taiwan, in 1988, the M.S. degree in information engineering from Tatung University, Taiwan, in 1990, and Ph.D. in information engineering from National Central University, Taiwan, in 1995. He was an assistant professor of Computer Information Science at Aletheia University, Taiwan, since August 1997. His current research interests include parallelizing compilers, interconnection networks, networks of workstation, and mobile computing.



Jang-Ping Sheu received the B.S. degree in computer science from Tamkang University, Taiwan, Republic of China, in 1981, and the M.S. and Ph.D. degrees in computer science from the National Tsing Hua University, Taiwan, Republic of China, in 1983 and 1987, respectively. He joined the faculty of the Department of Electrical Engineering, National Central University, Taiwan, Republic of China, as an Associate Professor in 1987. He is currently a Professor and Chair of the Department of Computer Science and Information Engineering, National Central University. From March to June of 1995, he was a visiting researcher at the IBM Thomas J. Watson Research Center, New York. His current research interests include parallelizing compilers, interconnection networks, and mobile computing. Dr. Sheu is a senior member of the IEEE, a member of the ACM and Phi Tau Phi Society. He is an associate editor of Journal of Information Science and Engineering, Journal of the Chinese Institute of Electrical Engineering, and Journal of the Chinese Institute of Engineers. He received the Distinguished Research Awards of the National Science Council of the Republic of China in 1993–1994, 1995–1996, and 1997–1998.