BROADCAST WORMHOLE-ROUTED 3-D MESH NETWORKS

Amnah El-Obaid

Department of Computer Science, Al-Zaytoonah University of Jordan, Amman, Jordan

ABSTRACT:

In a network, one-to-all broadcasting is the process of disseminating messages from a source node to all the nodes existing in the network through successive data transmissions between pairs of nodes. Broadcasting is the most primary communication process in a network. In this paper, we study on multiport wormhole-routed multicomputers where nodes are able to send multiple messages into the network at a time. We propose efficient broadcast algorithms in multi-port wormhole-routed multicomputers which are characterized by 3D mesh topology. The proposed algorithm Three-Dimension Broadcast Layers (3-DBl) is designed such that can send messages to destinations within two start-up communication phases for each 2-D mesh. The second proposed algorithm Three-Dimension Broadcast Surfaces (3-DBS) is designed such that can send messages to destinations within six start-up communication phases. The performance study in this paper clearly shows the advantage of the proposed algorithm.

Keywords:

Broadcast communication, Wormhole routing, 3-D mesh topology, Deadlock-free algorithm.

1. INTRODUCTION

Multicomputers are playing an increasing important role in scientific computation and high speed computing applications such as weather forecast and high performance servers. Multicomputers are usually organized as an ensemble of nodes, each of them equipped with its own processor, local memory, and other supporting devices. The nodes are interconnected using a variety of topologies. Most of the popular direct network topologies fall in the general category of either n-dimensional meshes or k-ary n-cubes because their regular topologies and simple routing. Among topologies, the most commonly used are meshes, tori, hypercubes and trees. The Mesh-based topologies are the most regular simple architecture that is used in multicomputers. Its implementation is very simple and easy to understand and verify. These are the soul properties which are necessary for any topology design [1]. Recent interest in multicomputer systems is therefore concentrated on two-dimensional or three-dimensional mesh and torus networks. Such technology has been adopted by the Intel Touchstone DELTA [2], MIT J-machine [3], Intel Paragon [4] and [5], Caltech MOSAIC [6], Cray T3D and T3E [7] and [8].

We consider the communication network using wormhole routing switching technology [9], [10] and [11], which is characterized with low communication latency. Wormhole flow control [12], also called wormhole switching or wormhole routing is a system of simple flow control in computer networking based on known fixed links. It is a subset of flow control methods called Flit-Buffer Flow Control. Large network packets are broken into small pieces called FLITs (flow control digits). The first flit, called the header flit holds information about this packet's route (namely the destination address) and sets up the routing behavior for all subsequent

DOI : 10.5121/ijcnc.2015.7411

flits associated with the packet. The head flit is followed by zero or more body flits, containing the actual pay load of data. The final flit, called the tail flit, performs some book keeping closing the connection between the two nodes. If the header encounters a channel already in use, it is blocked until the channel is freed [13], [14]. No communication can occur over the deadlocked channels until exceptional action is taken to break the deadlock. A multicomputer network is said to be in a deadlock condition when no message can advance towards its destination. This situation can postpone message delivery indefinitely. Deadlock can occur if messages are allowed to hold some resource while requesting others [15], [16]. Many algorithms have been proposed for broadcast communication in wormhole-routed networks over the past few years [17], [18] and [19].

In wormhole-routed networks, the communication latency is the most important performance metric in direct network; it consists of three parts, start-up latency network latency, and blocking latency. The start-up latency is the time required to start a message, which involves operation system overheads. The network latency consists of channel propagation and router latencies, i.e., the elapsed time after the head of a message has entered the network at the source until the tail of the message emerges from the network at the destination. The blocking latency accounts for all delays associated with contention for routing resources among the various worms in the network [20].

One of the most fundamental communication operations is broadcast, in which the same message is delivered from a source node to all nodes in the network. Efficient broadcast communication is useful in message-passing applications, and is also necessary in several other operations, such as replication and barrier synchronization [21], which are supported in data parallel languages. Many algorithms have been proposed for broadcast communication in wormhole-routed networks over the past few years [22], [23] [24], [25], [26], [27], [28] and [29]. A broadcasting algorithm is applicable in diversified manners in several fields such as management of shared variables for distributed programming, image processing, data copying in database of large-scale network, and data collection in wireless sensor network (WSN), and for this, an effective broadcasting algorithm is necessary [30].

This paper, discussion is restricted to the 3-D mesh topology with Bi-directional channels. The 3-D mesh topology can be modeled as a graph M (V, E) in which each node in V (M) corresponds to a processor and each edge in E (M) corresponds to a communication channel. The mesh graph is formally defined below. Where m (rows) x n (columns) x r (layers) 3-D mesh comprises mnr nodes interconnected in a grid fashion

Definition 1: An m x n x r non wraparound 3-D mesh graph is a directed graph M (V, E), where the following conditions exist:

$$
\begin{aligned}
&V(M) = \{(x, y, z) \mid 0 \le x < n, 0 \le y < m, 0 \le z < r\} \; and \\
&E(M) = \{[(x_i, y_i, z_i), (x_j, y_j, z_j)] \mid (x_i, y_i, z_i), (x_j, y_j, z_j) \in V(G), \\
&and \left| x_i - x_j \right| + \left| y_i - y_j \right| + \left| z_i - z_j \right| = 1\}
\end{aligned}
$$

In this paper we introduce the new broadcast algorithms based use wormhole routing for 3-D mesh multicomputers. The rest of the paper is organized as follows. Section 2 presents the related works. In section 3 we introduce the new broadcast algorithms 3-DBL and 3-DBS, while section 4 examines the performance of the proposed algorithms. Finally, section 5 concludes this study.

2. RELATED WORKS

Hamiltonian paths: A network partitioning strategy based on Hamiltonian paths is fundamental to the deadlock-free routing schemes. A Hamiltonian path visits every node in a graph exactly once [31]; a 2-D mesh has many Hamiltonian paths. In this algorithm, each node u in a multicomputer is assigned a label, L (u). In a network with N nodes, Hamiltonian path assigns a label for each node based on the position of that node in a mesh, where the first node in the path is labeled 0 and the last node in the path is labeled $N-1$. Fig. 1 shows such a labeling in a 4 x 4 mesh. The label assignment function L for an m x n mesh can be expressed in terms of the $x-$, $y-$ coordinates of nodes as follows:

$$L(x,y)=\begin{cases} y*n+x, & if\ y\, is\, even \\ y*n+n-x-1 & if\ y\, is\ odd \end{cases}$$

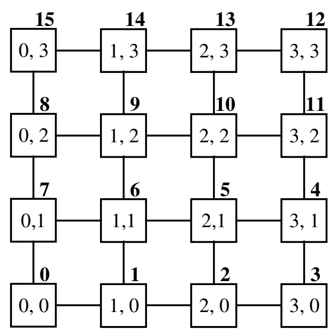

Fig. 1 The Hamiltonian labeling of 4 x 4 mesh

The labeling effectively divides the network into two subnetworks. The high-channel subnetwork contains all of the channels whose direction is from lower-labeled nodes to higher-labeled nodes, and the low-channel network contains all of the channels whose direction is from higher-labeled nodes to lower-labeled nodes. The source node divides the network into two subnetworks, N_U and N_L, where every node in N_U has a higher label than that of the source node and every node in N_L has a lower label than that of the source node. The message transmission is made according to the equation that presented in [31].

A source node divides the destination set D into two subsets, D_U and D_L, where D_U contain the destination nodes in N_U and D_L contain the destination nodes in N_L. The messages will be sent from the source node to the nodes in D_U using the high-channel network and to the destination nodes in D_L using the low-channel network.

Sort the destination nodes in D_U, using the L value as the key, in ascending order. Sort the destination nodes in D_L, using the L value as the key, in descending order. Construct two messages, one containing D_U as part of the header and the other containing D_L as part of the header. The source sends two messages into tow disjoint subnetworks N_U and N_L.

Consider the example shown in Fig. 2 for a 4 x 4 mesh topology labeling using a Hamiltonian path. The source node labeled 6 initiates a broadcast to the nodes. The source splits, and sorts, D in two subsets D_L = {5, 4, 3, 2, 1, 0} and D_U = {7, 8, 9, 10, 11, 12, 13, 14, 15}. The routing pattern is shown with bold and dashed lines in Fig. 2.

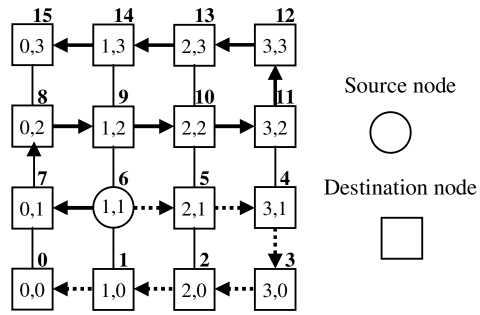

Fig. 2 The routing pattern of Hamiltonian path

3. THE PROPOSED ALGORITHMS

In this paper two algorithms are presented**.** Our first algorithm 3-DBL is based on splitting the 3-D mesh network into set of layers. Second algorithm 3-DBS is based on splitting the 3-D mesh network into set of surfaces (here groups of columns, rows and diagonals).

3.1 3-D Broadcast Layers (3-DBL)

Our first algorithm is based on splitting the 3-D mesh network into set of layers. Each layer will be labeled with Hamilton model for 2-D mesh as shown in [20]. The 3-D mesh network can be shown as set of layers; each layer represents a 2-D mesh network. In fact, if we have m x n x r 3-D mesh, then we have r layers of m x n 2-D mesh. Formally, the sub-networks are described by the following expression:

$$
\begin{aligned}
N_{Z0} &= \{(x, y, z) \mid (0 \le x < n), (0 \le y < m), (z = 0)\}, \\
N_{Z1} &= \{(x, y, z) \mid (0 \le x < n), (0 \le y < m), (z = 1)\}, \\
&\vdots \\
N_{Zr} &= \{(x, y, z) \mid (0 \le x < n), (0 \le y < m), (z = r-1)\}
\end{aligned}
$$

Suppose that the coordinate of the source node u_0 is represented by (x_0, y_0, z_0), and D represent the all nodes of 3-D mesh. The simple idea of this algorithm is as follow: -

Step 1: The source split the D into three subsets D_{Z0}, D_{ZU} and D_{ZL}, where D_{Z0} contain the nodes which their z coordinates are equal z_0, D_{ZU} contain the nodes which their z coordinates are greater than z_0 and D_{ZL} contain the nodes which their z coordinates are smaller than z_0. Formally, the subsets are described by the following expression:

$$D_{Z0} = \{(x, y, z) | (x, y, z) \in D, (0 \le x < n), (0 \le y < m), (z = z_0)\},$$
$$D_{ZU} = \{(x, y, z) | (x, y, z) \in D, (0 \le x < n), (0 \le y < m), (z_0 < z < r)\},$$
$$D_{ZL} = \{(x, y, z) | (x, y, z) \in D, (0 \le x \prec n), (0 \le y \prec m), (0 \le z < z_0)\}$$

Step 2: The source sends the message to all nodes in D_{Z0} using the Hamilton model for 2-D mesh, which is explained above.
Step 3: Divide D_{ZU} into n subsets, D_{ZU1}, $D_{ZU2}, \ldots\ldots\ldots$ D_{ZUN}. Formally, the subsets are described by the following expression:

$$D_{ZU1} = \{(x, y, z) | (x, y, z) \in D_{ZU}, (0 \le x < n), (0 \le y < m), (z = z_0 + 1)\},$$
$$D_{ZU2} = \{(x, y, z) | (x, y, z) \in D_{ZU}, (0 \le x < n), (0 \le y < m), (z = z_0 + 2\},$$

.

.

.

$$D_{ZUN} = \{(x, y, z) | (x, y, z) \in D_{ZU}, (x \le 0 < n), (0 \le y < m), (z = r - 1\}$$

Step 4: Each above subset represents a layer of 2-D mesh. The source sends the message to a node, which is represented by its integer coordinate (x_0, y_0, z_{DZU1}) which will be act as a source in its layer, to a node represented by its integer coordinate (x_0, y_0, z_{DZU2}) which will be act as a source in its layer and so on to a node represented by its integer coordinate (x_0, y_0, z_{DZUN}) which will be act as a source in its layer.

Step 5: The alternative source (x_0, y_0, z_{DZU1}) will sends the message to all nodes in D_{ZU1} using Hamilton model of 2-D mesh, the alternative source (x_0, y_0, z_{DZU2}) will sends the message to all nodes in D_{ZU2} using Hamilton model of 2-D mesh and so on.

Step 6: Divide D_{ZL} into n subsets, D_{ZL1}, $D_{ZL2}, \ldots\ldots\ldots$ D_{ZLN}. Formally, the subsets are described by the following expression:

$$D_{ZL1} = \{(x, y, z) | (x, y, z) \in D_{ZL}, (0 \le x < n), (0 \le y < m), (z = 0)\},$$
$$D_{ZL2} = \{(x, y, z) | (x, y, z) \in D_{ZL}, (0 \le x < n), (0 \le y < m), (z = 1)\},$$

.

.

.

$$D_{ZLN} = \{(x, y, z) | (x, y, z) \in D_{ZL}, (x \le 0 < n), (0 \le y < m), (z = z_0 - 1)\}$$

Step 7: Repeat step 4 and step 5 for each nodes in D_{ZL1}, $D_{ZL2}, \ldots\ldots\ldots$ D_{ZLN},

3.1.1 Comparative study

To demonstrate the performance of 3-DBL algorithm, consider the example shown in Fig. 3 for a 4 x 4 x 4 mesh topology labeling using a Hamiltonian path for each 2-D mesh. The source node labeled 6 with integer coordinate (1, 1, 1) initiates a broadcast to all nodes D in the 3-D mesh.

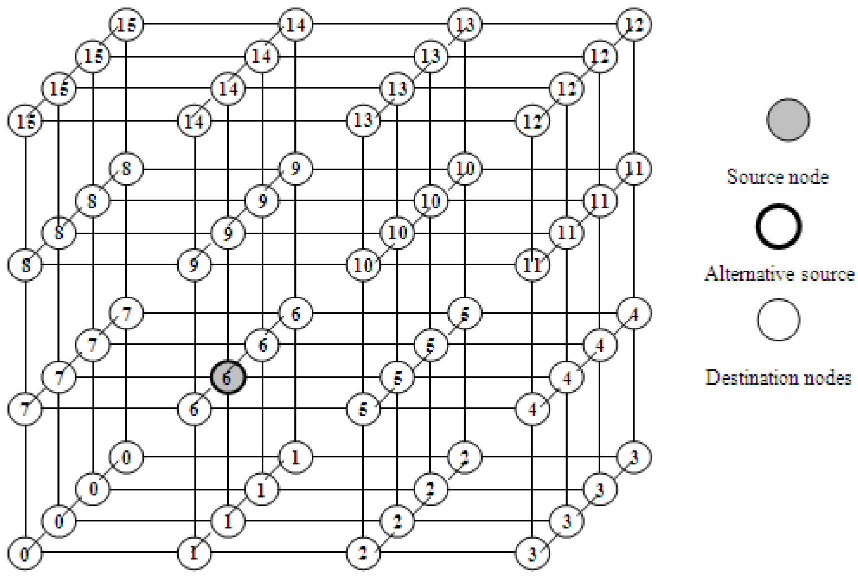

Fig. 3 An example of 4x4x4 mesh

The 3-DBL algorithm splits D into three subsets D_{Z0}, D_{ZU} and D_{ZL} as follows:-

D_{Z0} = {(0,0,1), (1,0,1), (2,0,1), (3,0,1), (0,1,1), (2,1,1), (3,1,1), (0,2,1), (1,2,1), (2,2,1), (3,2,1), (0,3,1), (1,3,1), (2,3,1), (3,3,1) }
D_{ZU} = { (0,0,2), ((1,0,2), (2,0,2), (3,0,2), (0,1,2), (1,1,2), (2,1,2), (3,1,2), (0,2,2), (1,2,2), (2,2,2), (3,2,2), (0,3,2), (1,3,2), (2,3,2), (3,3,2), (0,0,3), (1,0,3), (2,0,3), (3,0,3), (0,1,3), (1,1,3), (2,1,3), (3,1,3), (0,2,3), (1,2,3), (2,2,3), (3,2,3), (0,3,3), (1,3,3), (2,3,3)(3,3,3) }

D_{ZL} = { (0,0,0), (1,0,0), (2,0,0), (3,0,0), (0,1,0), (1,1,0), (2,1,0), (3,1,0), (0,2,0), (1,2,0), (2,2,0), (3,2,0), (0,3,0), (1,3,0), (2,3,0), (3,3,0) }

The 3-DBL divide D_{ZU} into two subsets D_{ZU1} and $D_{ZU2,}$ as follows:

D_{ZU1} = { (0,0,2), ((1,0,2), (2,0,2), (3,0,2), (0,1,2), (1,1,2), (2,1,2), (3,1,2), (0,2,2), (1,2,2), (2,2,2), (3,2,2), (0,3,2), (1,3,2), (2,3,2), (3,3,2) }

D_{ZU2} = { (0,0,3), (1,0,3), (2,0,3), (3,0,3), (0,1,3), (1,1,3), (2,1,3), (3,1,3), (0,2,3), (1,2,3), (2,2,3), (3,2,3), (0,3,3), (1,3,3), (2,3,3)(3,3,3) }

The routing pattern of 3-DBL algorithm is shown with bold and dashed lines in Fig. 4(a) for 2-D mesh where z coordinate =1, in Fig. 4(b) for 2-D mesh where z coordinate =2, in Fig. 4(c) for 2-D mesh where z coordinate =3 and in Fig. 4(d) for 2-D mesh where z coordinate =0.

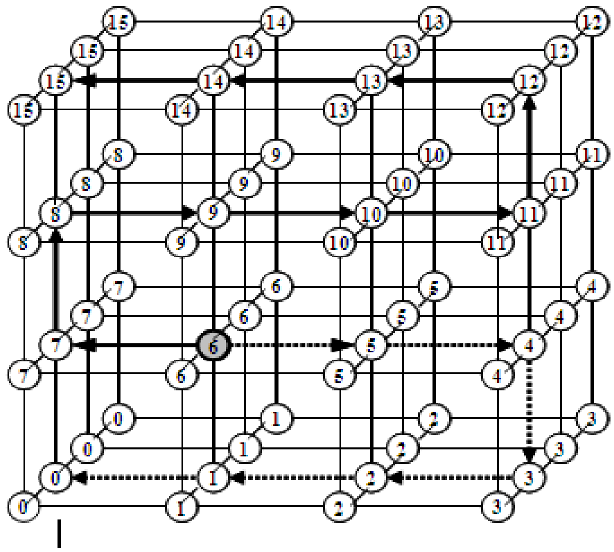

Fig. 4 (a). The routing patterns of 3-DBL algorithm, for 2-D mesh where z coordinate = 1

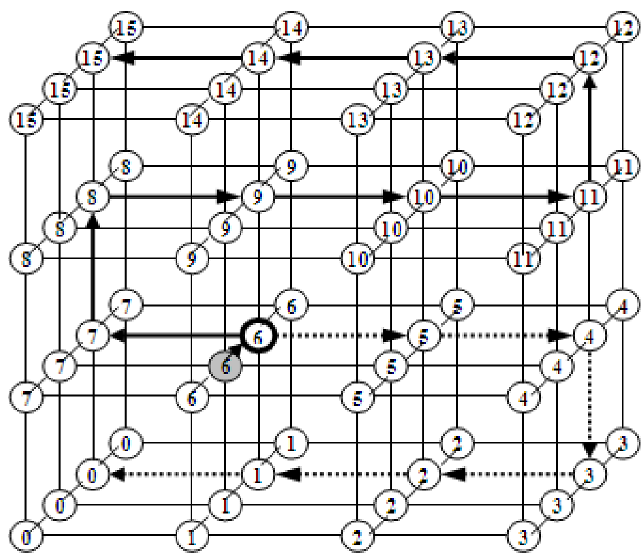

Fig. 4 (b). The routing patterns of 3-DBL algorithm, for 2-D mesh where z coordinate = 2

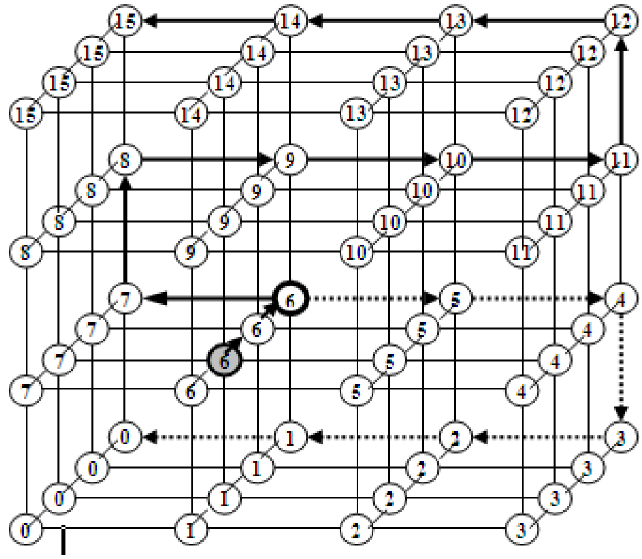

Fig. 4 (c). The routing patterns of 3-DBL algorithm, for 2-D mesh where z coordinate = 3

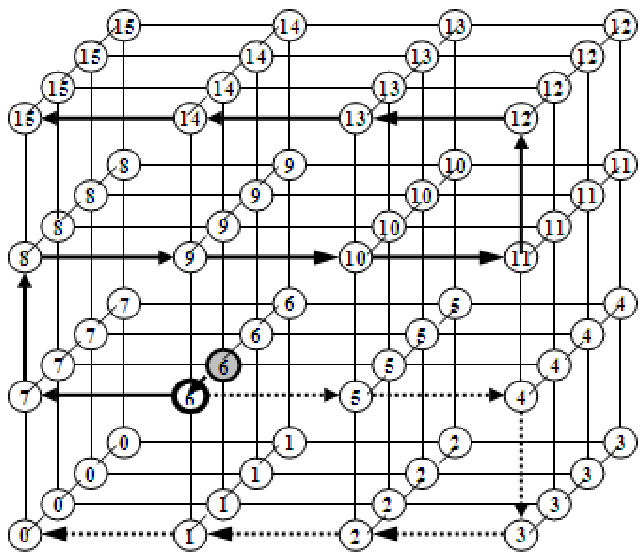

Fig. 4 (d). The routing patterns of 3-DBL algorithm, for 2-D mesh where z coordinate = 0

Theorem 1: 3-DBL algorithm is deadlock-free.

Proof: For m x n x r 3-D network, the source node divides the network into r disjoint subnetworks N_{Z0}, N_{Z1},….N_{Zr}. This is obvious since, $N_{Z0} \cap N_{Z1} \cap \cdots N_{Zr} = \phi$. Then 3-DBL algorithm is deadlock-free at r subnetworks. Now, we will prove that there are no dependencies within each subnetwork. The 3-DBL algorithm uses Hamiltonian-path routing to distribute destination nodes for each subnetwork, as proved in [20] Hamiltonian-path algorithm is deadlock-free; therefore, no cyclic dependency can exist among the channels. Hence 3-DBL algorithm is deadlock-free.

3.2 3-D Broadcast Surfaces (3-DBS)

Our second proposed algorithm based on splitting the network into six subnetworks, N_{XR}, N_{XL}, N_{YU}, N_{YL} , N_{ZU} and N_{ZL}. Subnetwork N_{XR} contains all right horizontal channels of source, subnetwork N_{XL} contains all Left horizontal channels of source, subnetwork N_{YU} contains all upper vertical channels of source, subnetworks N_{YL} contains all Lower vertical channels of source, subnetworks N_{ZU} contains all upper diagonal channels of source and subnetworks N_{ZL} contains all lower diagonal channels of source. Suppose that the coordinate of the source node u_0 is represented by (x_0, y_0, z_0), and D represents the all nodes in 3-D mesh, the simple idea of this algorithm is as follow: -

Step 1: The source split the D into three subsets, D_{XS}, D_{XR} and D_{XL}. Formally, the subsets are described by the following expression:

$$D_{XS} = \{(x, y, z) | (x, y, z) \in D, x = x_0, (0 \le y < m), (0 \le z < r)\},$$
$$D_{XR} = \{(x, y, z) | (x, y, z) \in D, x > x_0, (0 \le y < m), (0 \le z < r)\},$$
$$D_{XL} = \{(x, y, z) | (x, y, z) \in D, x < x_0, (0 \le y < m), (0 \le z < r)\}$$

Step 2: The source split D_{XS} into four subsets D_{ZU}, D_{ZL}, D_{YU} and D_{YL}. Formally, the subsets are described by the following expression:

$$D_{ZU} = \{(x, y, z) | (x, y, z) \in D_{XS}, x = x_0, y = y_0, z > z_0\},$$
$$D_{ZL} = \{(x, y, z) | (x, y, z) \in D_{XS}, x = x_0, y = y_0, z < z_0\},$$
$$D_{YU} = \{(x, y, z) | (x, y, z) \in D_{XS}, x = x_0, y > y_0, (0 \le z < r)\},$$
$$D_{YL} = \{(x, y, z) | (x, y, z) \in D_{XS}, x = x_0, y < y_0, (0 \le z < r)\}$$

Step 3: Sort subset D_{ZU} using the z coordinate as the key in ascending order, sort subset D_{ZL} using the z coordinate as the key in descending order, sort subset D_{YU} using the y coordinate as the key in ascending order and sort subset D_{YL} using the y coordinate as the key in descending order.

Step 4: The source sends the message to above subsets as follows:-

1. Sends to subset DXR through subnetwork NXR, to next right node of u0, which is represented by integer coordinate (x0+1, y0, z0).
2. Sends to subset DXL through subnetwork NXL, to next left node of u0, which is represented by integer coordinate (x0-1, y0, z0).
3. Sends to subset DYU through subnetwork NYU, to next upper vertical node of u0, which is represented by integer coordinate (x0, y0+1, z0).
4. Sends to subset DYL through subnetwork NYL, to next lower vertical node of u0, which is represented by integer coordinate (x0, y0-1, z0).
5. Sends to subset DZU through subnetwork NZU to next upper diagonal node of u0, which is represented by integer coordinate (x0, y0, z0+1) and so on until reach all nodes in upper diagonal.
6. Sends to subset DZL through subnetwork NZL, to next lower diagonal node of u0, which is represented by integer coordinate (x0, y0, z0-1) and so on until reach all nodes in lower diagonal.

The source would simultaneously send these messages by six ports (3-DBS algorithm is all ports). In this step all diagonal subsets of source will be reached.

Step 5: The intermediate node that represented by (x_0, y_0+1, z_0) will be act as a source on its surface, when it receives the subset $D_{YU,}$ it will split it as follows:-

$$D_{YU} \begin{cases} \rightarrow A=\{(x,y,z) \mid (x,y,z)\in D_{YU}, x=x_0, (y=y_0+1), (0\le z<r)\}, \\ \rightarrow B=\{(x,y,z) \mid (x,y,z)\in D_{YU}, x=x_0, (y=y_0+2), (0\le z<r)\}, \\ \vdots \\ \rightarrow N=\{(x,y,z) \mid (x,y,z)\in D_{YU}, x=x_0, (y=m-1), (0\le z<r)\} \end{cases}$$

$$A \begin{cases} \rightarrow A_{ZU}=\{(x,y,z) \mid (x,y,z)\in A, x=x_0, (y=y_0+1), z>z_0\}, \\ \rightarrow A_{ZL}=\{(x,y,z) \mid (x,y,z)\in A, x=x_0, (y=y_0+1), z<z_0\} \end{cases}$$

$$B \begin{cases} \rightarrow B_{ZU}=\{(x,y,z) \mid (x,y,z)\in B, x=x_0, (y=y_0+2), z>z_0\}, \\ \rightarrow B_{ZL}=\{(x,y,z) \mid (x,y,z)\in B, x=x_0, (y=y_0+2), z<z_0\} \end{cases}$$

$$\vdots$$

$$N \begin{cases} \rightarrow N_{ZU}=\{(x,y,z) \mid (x,y,z)\in N, x=x_0, (y=m-1), z>z_0\}, \\ \rightarrow N_{ZL}=\{(x,y,z) \mid (x,y,z)\in N, x=x_0, (y=m-1), z<z_0\} \end{cases}$$

The source sends the messages to subsets $A_{ZU,}$ B_{ZU} and so on to N_{ZU} until reach all nodes in upper diagonals. The source sends the messages to subsets $A_{ZL,}$ B_{ZL} and so on to N_{ZL} until reach all nodes in lower diagonals.

Step 6: The intermediate node that represented by (x_0, y_0-1, z_0) will be act as a source on its surface, when it receives the destination subset D_{YL}, it will spilt it as follows and apply the same partition schema that presented in step5 for subsets A, B ….N.

$$D_{YL} \begin{cases} \rightarrow A=\{(x,y,z) \mid (x,y,z)\in D_{YL}, x=x_0, (y=0), (0\le z<r)\}, \\ \rightarrow B=\{(x,y,z) \mid (x,y,z)\in D_{YL}, x=x_0, (y=1), (0\le z<r)\}, \\ \vdots \\ \rightarrow N=\{(x,y,z) \mid (x,y,z)\in D_{YL}, x=x_0, (y=z_0-1), (0\le z<r)\} \end{cases}$$

Step 7: The intermediate node that represented by (x_0+1, y_0, z_0) will be act as a source on its surface, when it receives the destination subset D_{XR}, it will spilt D_{XR} as follows:-

$$D_{XR} \begin{cases} \rightarrow A=\{(x,y,z) \mid (x,y,z)\in D_{XR}, (x=x_0+1), (0\le y<m), (0\le z<r)\}, \\ \rightarrow B=\{(x,y,z) \mid (x,y,z)\in D_{XR}, (x=x_0+2), (0\le y<m), (0\le z<r)\}, \\ \vdots \\ \rightarrow N=\{(x,y,z) \mid (x,y,z)\in D_{XR}, (x=n-1), (0\le y<m), (0\le z<r)\} \end{cases}$$

$$A \rightarrow \begin{cases} A_{YU} = \{ (x, y, z) \mid (x, y, z) \in A, (x = x_0 + 1), y > y_0, (0 \le z < r) \}, \\ A_{YL} = \{ (x, y, z) \mid (x, y, z) \in A, (x = x_0 + 1), y < y_0, (0 \le z < r) \} \end{cases}$$

:

$$B \rightarrow \begin{cases} B_{YU} = \{ (x, y, z) \mid (x, y, z) \in B, (x = x_0 + 2), y > y_0, (0 \le z < r) \}, \\ B_{YL} = \{ (x, y, z) \mid (x, y, z) \in B, (x = x_0 + 2), y < y_0, (0 \le z < r) \} \end{cases}$$

$$N \rightarrow \begin{cases} N_{YU} = \{ (x, y, z) \mid (x, y, z) \in N, (x = n - 1), y > y_0, (0 \le z < r) \}, \\ N_{YL} = \{ (x, y, z) \mid (x, y, z) \in N, (x = n - 1), y < y_0, (0 \le z < r) \} \end{cases}$$

Repeat the previous step 5 and 6.

Step 8: The intermediate node that represented by (x_0-1, y_0, z_0) will be act as a source on its surface, when it receives the destination subsets D_{XL}, it will spilt it as follows and apply the same partition schema that presented in step7 for subsets A, B ….N.

$$D_{XL} \rightarrow \begin{cases} A = \{ (x, y, z) \mid (x, y, z) \in D_{XL}, (x = 0), (0 \le y < m), (0 \le z < r) \}, \\ B = \{ (x, y, z) \mid (x, y, z) \in D_{XL}, (x = 1), (0 \le y < m), (0 \le z < r) \}, \\ \vdots \\ N = \{ (x, y, z) \mid (x, y, z) \in D_{XL}, (x = x_0 - 1), (0 \le y < m), (0 \le z < r) \} \end{cases}$$

3.2.1 Comparative study

To demonstrate the performance of 3-DBS algorithm, consider the example shown in Fig. 5 for a 4 x 4 x 4 mesh topology. The source node with integer coordinate (1, 1, 1) initiates a broadcast to all D nodes in the 3-D mesh. The source dive the D into the following:-

D_{XS} = { (1,0,0), (1,0,1), (1,0,2), (1,0,3), (1,1,0), (1,1,2), (1,1,3), (1,2,0) (1,2,1), (1,2,2), (1,2,3), (1,3,0), (1,3,1), (1,3,2), (1,3,3) }

D_{XR} = { (2,0,0), (2,0,1), (2,0,2), (2,0,3), (2,1,0), (2,1,1), (2,1,2), (2,1,3), (2,2,0), (2,2,1), (2,2,2), (2,2,3), (2,3,0), (2,3,1), (2,3,2), (2,3,3), (3,0,0), (3,0,1), (3,0,2), (3,0,3), (3,1,0), (3,1,1), (3,1,2), (3,1,3), (3,2,0), (3,2,1), (3,2,2), (3,2,3), (3,3,0), (3,3,1), (3,3,2), (3,3,3) }

D_{Xl} = { (0,0,0), (0,0,1), (0,0,2), (0,0,3), (0,1,0), (0,1,1), (0,1,2), (0,1,3), (0,2,0), (0,2,1), (0,2,2), (0,2,3), (0,3,0), (0,3,1), (0,3,2), (0,3,3) }

The source dived D_{XS} into D_{ZU}, D_{ZL}, D_{YU} and D_{YL} as follows:-

D_{ZU} = {(1,1,2), (1,1,3) }
D_{ZL} = { (1,1,0) }
D_{YU} = { (1,2,0), (1,2,1), (1,2,2), (1,2,3), (1,3,0), (1,3,1), (1,3,2), (1,3,3) }
D_{YL} = { ((1,0,0), (1,0,1), (1,0,2), (1,0,3) }

The source repeats this partition strategy for D_{XR} and D_{XL}. The routing pattern of 3-DBS algorithm is shown with bold lines in Fig. 5.

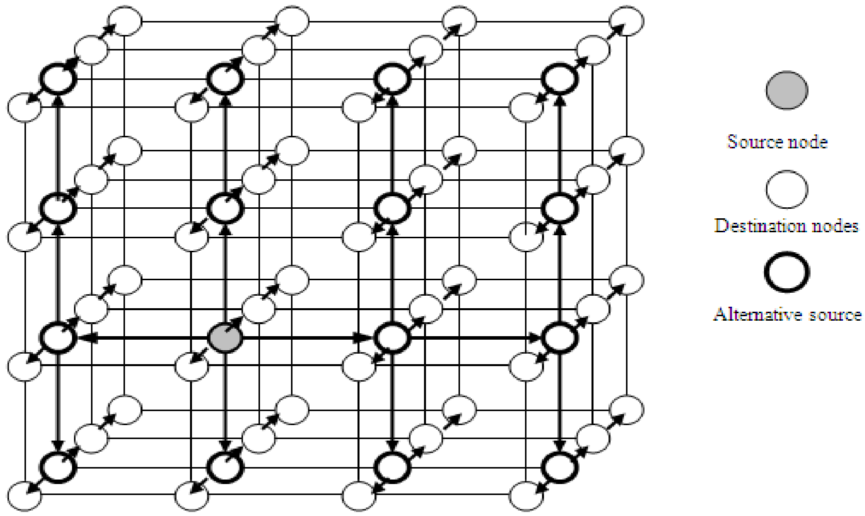

Fig. 5 The routing patterns of 3-DBS algorithm

Theorem 2: The 3-DBS algorithm is deadlock-free.

Proof: At the source node, 3-DBS algorithm divides the network into six disjoint subnetworks. This is obvious since, $N_{ZU} \cap N_{ZL} \cap N_{YU} \cap N_{YL} \cap {}_{NXR} \cap N_{XL} = \varphi$. Then 3-DBS algorithm is deadlock-free at the six subnetworks. Now, we will prove that there are no dependencies within each subnetwork. In subnetwork N_{ZU}, using the 3-DBS algorithm, a message entering a node with coordinate (x, y, z) always leaves on a node with coordinate (x, y, z+1); therefore, no cyclic dependency can exist among the channels. In subnetwork N_{ZL}, using the 3-DBS algorithm, a message entering a node with coordinate (x, y, z) always leaves on a node with coordinate (x, y, z-1); therefore, no cyclic dependency can exist among the channels. In subnetwork N_{YU}, using the 3-DBS algorithm, a message entering a node with coordinate (x, y, z) always leaves on a node with coordinate (x, y+1, z); therefore, no cyclic dependency can exist among the channels. In subnetwork N_{YL}, using the 3-DBS algorithm, a message entering a node with coordinate (x, y, z) always leaves on a node with coordinate (x, y-1, z); therefore, no cyclic dependency can exist among the channels. In subnetwork N_{XR}, using the 3-DBS algorithm, a message entering a node with coordinate (x, y, z) always leaves on a node with coordinate (x+1, y, z); therefore, no cyclic dependency can exist among the channels. In subnetwork N_{XL}, using the 3-DBS algorithm, a message entering a node with coordinate (x, y, z) always leaves on a node with coordinate (x-1, y, z); therefore, no cyclic dependency can exist among the channels. Hence 3-DBL algorithm is deadlock-free.

4. SIMULATION

In order to compare the performance of our proposed broadcast routing algorithms, the simulation program used to model broadcast communication in 3-D mesh networks is written in VC++ and uses an event-driven simulation package, CSIM [32]. CSIM allows multiple processes to execute in a quasiparallel fashion and provides a very convenient interface for writing modular simulation programs. The simulation program for broadcast communication is part of a larger simulator, called MultiSim [33]. MultiSim was used to simulate broadcast operations in 3D mesh of different sizes. For a given message length, a large number of different source nodes were selected at random to perform the broadcast operation. All simulations were performed for a 5 x 5 x 5 3-D mesh.

We examined the routing performance of our proposed algorithms under various, startup latencies ß, and message lengths. The message length is the number of flits in a message and the message startup latency includes the software overhead for buffers allocating, messages coping, router initializing, etc. The maximum latencies were measured and then averaged over these samples.

Figure 6 compares the two algorithms across different message lengths (100 bytes to 2000 bytes) in 5 x 5 x 5 network. In Fig 6(a), the startup latency ß is set to 10 microseconds. In Fig 6(b), the startup latency ß is set to 100 microseconds. In both cases, the advantage of the 3-DBS algorithm is significant. The 3-DBS algorithm is less sensitive rather than the 3-DBL algorithm, however, the disadvantage of 3-DBL algorithm increases with the message lengths.

In the 3-DBS algorithm, destinations are divided into multiple sets (number of columns, rows, and diagonals). A source node may send a message through six ports simultaneously without channel contentions. Thus, the 3-DBS algorithm can efficiently deliver a broadcast message in terms of the message passing steps without stepwise contention from any intermediate source node. In 3-DBL algorithm 3-D mesh are divided into r layers of 2-D mesh and each layer divided their destinations into two subsets (use Hamilton-Path for each 2-D mesh). Since the number of messages waiting to be delivered in the node's memory depends mainly on the number of ports provided in the wormhole-routed multicomputer system, the system providing a large number of ports is insensitive to the message lengths.

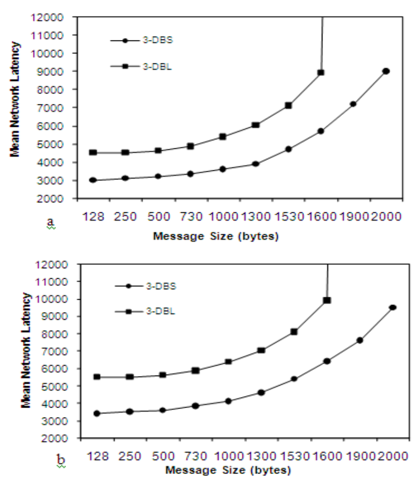

Fig. 6 Comparison of 3-DBL with 3-DBS. (a) With small message latency ß =10; (b) with large message latency ß =100

5. CONCLUSION

Broadcast, as one of the most fundamental collective communications operations, is highly demanded in parallel applications that are implemented in massively parallel computers. In this paper, new two broadcast algorithms in 3-D mesh parallel machines using wormhole facility were presented. These algorithms are shown to be deadlock-free. The behavior of 3-DBS algorithm was compared with behavior of 3-DBL algorithm using simulation. The result of the simulation study shows that the 3-DBS algorithm achieves better performance because it’s designed specifically to exploit the all-port architecture.

6. FUTURE WORK

There are several aspects of the mesh topology that need to be studied or expanded in the future. Performance of proposed algorithms that presented in this paper can be expanded to apply in hypercubes and general N-D meshes

ACKNOWLEDGEMENTS

We would like to thank the anonymous reviewers for their suggestions and constructive criticism.

REFERENCES

[1] Mohammad Yahiya Khan, Sapna Tyagi And Mohammad Ayoub Khan, (2014) "Tree-Based 3-D Topology For Network-On-Chipworld", Applied Sciences Journal, Vol. 30, No.7, Pp 844-851.
[2] Intel Corporation, (1990) "A Touchstone Delta System Description", Intel Corporatio, Intel Supercomputing Systems Division.
[3] Nuth P. R., And Dally W. J., (1992) "The J-Machine Network", In Proc. Ieee Int. Conf. On Computer Design: Vlsi In Computer And Processors, Pp 420-423, Ieee Computer Society Press.
[4] R. Foschia, T. Rauber, And G. Runger, (1997) "Modeling The Communication Behavior Of The Intel Paragon, In Modeling, Analysis, And Simulation Of Computer And Telecommunication Systems", Ieee Computer Society Press, Pp 117-124.
[5] G. S. Almasi, And A. Gottlieb, (1994) "Highly Parallel Computing Benjamin/Cummings"
[6] W. C. Athas, And C. L. Seitz, (1988) "Multicomputers Message Passing Concurrent Computers", Ieee Comp, Vol. 21, No. 8, Pp 9-24.
[7] R. E. Lessler, And J. L. Schwazmeier, (1993) "Cray T3d: A New Dimension For Cray Research In Compcon", Ieee Computer Society Press, Pp 176-182.
[8] Cray Research Inc, (1995) "Cray T3e Scalable Parallel Processing System", Cray Research Inc.,.Http://Www.Cray.Com/Products/Systems/Crayt3e/.
[9] Y.-C. Tseng, D.K. Panda, And T.-H. Lai, Feb. (1996) "A Trip-Based Multicasting Model In Wormhole-Routed Networks With Virtual Channels", Ieee Trans. Parallel And Distributed Systems, Vol. 7, No. 2, Pp 138-150.
[10] W. Dally And C. Seitz, (1986) "The Torus Routing Chip, J. Distributed Computing", Vol. 1, No. 3, Pp 187-19.
[11] L.M. Ni And P.K. Mckinley, Feb. (1993) "A Survey Of Wormhole Routing Techniques In Directed Network, Computer", Vol. 26, No. 2, Pp 62-76.
[12] William James Dally; Brian Towles, (2004) "Principles And Practices Of Interconnection Networks. Morgan Kaufmann Publishers", Vol. 13, No. 2, Isbn 978-0-12-200751-4.
[13] H. Moharam, M. A. Abd El-Baky, And S. M. M., October (2000) "Yomna- An Efficient Deadlock-Free Multicast Wormhole Algorithm In 2-D Mesh Multicomputers", Journal Of Systems Architecture, Vol. 46, No. 12, Pp 1073-1091.
[14] Nen-Chung Wang, Chih-Ping Chu, And Tzung-Shi Chen, (2002) "A Dual Hamiltonian-Path-Based Multicasting Strategy For Wormhole Routed Star Graph Interconnection Networks", J. Parallel Distrib. Comput. Vol. 62, Pp1747–1762.
[15] S. Taktak, E. Encrenaz, And J.-L. Feb. (2010) "Desbarbieux, A Polynomial Algorithm To Prove Deadlock-Freeness Of Wormhole Networks", Proc. 18th Euromicro Intl Conf. Parallel, Distributed And Network-Based Computing (Pdp '10).
[16] Verbeek And J. Schmaltz, Oct. (2011) "A Comment On A Necessary And Sufficient Condition For Deadlock-Free Adaptive Routing In Wormhole Networks", Ieee Trans. Parallel And Distributed Systems, Vol. 22, No. 10, Pp 1775-1776.
[17] Mohammad Yahiya Khan, Sapna Tyagi And Mohammad Ayoub Khan, (2014) "Tree-Based 3-D Topology For Network-On-Chipworld", Applied Sciences Journal, Vol. 30, No.7, Pp 844-851.
[18] Faizal Arya Samman, (2011) "New Theory For Deadlock-Free Multicast Routing In Wormhole Switched Virtual Chanel Less Networks On-Chip", Ieee Transactions On Parallel & Distrbuted System, Vol. 22, Pp 544-557.

[19] Mahmoud Omari, (2014) "Adaptive Algorithms For Wormhole-Routed Single-Port Mesh Hypercube Network", Jcsi International Journal Of Computer Science Issues, Vol. 11, No 1, Pp 1694-0814.
[20] Kadry Hamed, Mohamed A. El-Sayed, Btl, (2015) "An Efficient Deadlock-Free Multicast Wormhole Algorithm To Optimize Traffic In 2d Torus Multicomputer", International Journal Of Computer Applications, Vol. 111, No 6, Pp 0975 – 8887.
[21] Wang Hao & Wu Ling, (2012) "Preconcerted Wormhole Routing Algorithm For Mesh Structure Based On The Network On Chip", Information Management, Innovation Management And Industrial Engineering (Iciii), International Conference, Vol. 2, No. 2, Pp 154 – 158.
[22] Mahmoud Moadeli And Wim Vanderbauwhede, (2009) "A Communication Model Of Broadcast In Wormhole-Routed Networks On-Chip", International Conference On Advanced Information Networking And Applications.
[23] Jung-Hyun Seo & Hyeongok Lee, (2013) "Link-Disjoint Broadcasting Algorithm In Wormhole-Routed 3d Petersen-Torus Networks", International Journal Of Distributed Sensor Networks, Vol. 2013 , No. 501974, 7 Pages
[24] Z. Shen, (2007) " A Generalized Broadcasting Schema For The Mesh Structures", Applied Mathematics And Computation, Vol. 186, No. 2, Pp 1293–1310.
[25] J.-H. Seo, (2013) "Three-Dimensional Petersen-Torus Network: A Fixed-Degree Network For Massively Parallel Computers", Journal Of Supercomputing, Vol. 64, No. 3, Pp 987–1007.
[26] Li, Yamin, Shietung Peng, & Wanming Chu, (2012) "Hierarchical Dual-Net: A Flexible Interconnection Network And Its Routing Algorithm", International Journal Of Networking And Computing, Vol. 2, No. 2, Pp 234–250.
[27] V. Anand, N. Sairam And M. Thiyagarajan, (2012) "A Review Of Routing In Ad Hoc Networks" , Research Journal Of Applied Sciences, Engineering And Technology Vol. 4, No. 8, Pp 981-986.
[28] Jung-Hyun Seo And Hyeongok Lee, (2013) "Link-Disjoint Broadcasting Algorithm In Wormhole-Routed 3d Petersen-Torus Networks", International Journal Of Distributed Sensor Networks, Vol. 2013, Article Id 501974, 7 Pages, .
[29] Yuh-Shyan Chen And Yuan-Chun Lin, (2001) "A Broadcast-Vod Protocol In An Integrated Wireless Mobile Network", Journal Of Internet Technology, Vol. 2, No. 2, Pp. 143-154, (2010 Sci If=0.448).
[30] Mahmoud Moadeli And Wim Vanderbauwhede, (2009) "A Communication Model Of Broadcast In Wormhole-Routed Networks On-Chip", International Conference On Advanced Information,.
[31] Lin X., Mckinley P.K., Ni L.M., August (1994) "Deadlock-Free Multicast Wormhole Routing In 2-D Mesh Multicomputers", Ieee Trans. On Parallel And Distrib. Syst. Vol. 5, No. 8, Pp 793-804,
[32] Schwetman H. D., (1985) "Csim: A C-Based, Process-Oriented Simulation Language", Tech. Rep. Pp.080.85, Microelectronics And Computer Technology Corp.
[33] Mckinley P. K. And Trefftz C., (1993) "Multisim: A Tool For The Study Of Large-Scale Multiprocessors", In Proc. Int. Workshop On Modeling, Analysis And Simulation Of Computer And Telecommun. Nehvorks (Mascots 93), Pp. 57-62.

AUTHORS

Amnah Elobaid received her B.Sc. degree in Computer Sciences from Faculty of Science, Kink Abdul Aziz University, Sudia Arabia in 1993. She received her M.Sc. degree in Computer Science from Al Al-Bayt University, Amman, Jordan in 2000. From 2000 to 2006, she was a lecturer in the Department of Computer Science at Applied Science University, Amman, Jordan. She received a PhD degree in Computer Science from Jilin University, Changchun, China. Now she is Assistant Professor in Department of Computer Science, Faculty of Science and Information Technology, Al-Zaytoonah University of Jordan, Amman, Jordan. Her research interests include parallel processing, message-passing systems.