Deadlock Free 3-D Hamiltonian Broadcast Two-Phase Multi-Port Algorithm

Amnah El-Obaid

Department of Basic Science, Faculty of Science and Information Technology, Al-Zaytoonah University of Jordan, Amman, Jordan

Mailto: dramna@zuj.edu.jo

Abstract: Broadcast is one of the most important approach in distributed memory parallel computers that is used to find a routing approach from a one source to all nodes in the mesh. The approach of this case of Broadcasting is to send the message from one source to all destinations in the network which corresponds to one-to-all communication. Routing schema is the approach that used to determine the road that is used to send a message from a source node to destination nodes. Wormhole routing is a fundamental routing mechanism in modern parallel computers which is characterized with low communication latency. We show how to apply this approach to 3-D meshes. In wormhole routing large network packets are broken into small pieces called FLITs (flow control digits). The destination address is kept in the first flit which is called the header flit and sets up the routing behavior for all subsequent flits associated with the packet. If the packets of the message can't deliver to their destination and there is a cyclic dependence over the channels in the network, then the deadlock even is occurred. In this paper, we consider an efficient algorithm for broadcasting on Multi-Port wormhole-routed 3D mesh with arbitrary size. We introduce an efficient algorithm, Three-Dimension Tow Phase Multi-Port (3-DTPMP) algorithm which used broadcast communication facility with deadlock-free wormhole routing in general three-dimensional networks. In this paper the behaviors of this algorithm were compared to the previous results, our paradigm reduces broadcast latency and is simpler. In this paper our simulation results show the average of our proposed algorithm over the other algorithms that presented

Keywords: Broadcasting communication, Wormhole routing, Hamiltonian model, 3-D mesh, Deadlock-free

1. INTRODUCTION

One of the important and popular architecture in both of scientific computation and high speed computing applications is multicomputer architecture. In a multicomputer network, processors often need to communicate with each other for various reasons, such as data exchange or event synchronization. Efficient communication has been recognized to be critical for high performance computing. Multicomputer network consists of hundreds or thousands of nodes connected in some fixed topology; each of them contains a microprocessor, local memory, and other supporting devices. 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. This is because that the implementation of mesh is very simple and easy to understand. These properties which are necessary for every 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].

The way of the message that needed to visit the destination nodes is called the switching method. One of the efficient message routing algorithms of multicomputers is wormhole-switching. The wormhole switching is considered widely used in routing algorithms because of some reasons, firstly the buffering that is required in it is low buffering, allowing for efficient router implementation [9]. Secondly, it characterized with low communication latency [10].

In wormhole-routed networks, a message is divided into packets; the packets are divided into flits. A flit is the smallest unit of information that a channel can accept or refuse [11]. All routing information of the message is kept in the header flit(s) and the data elements are kept in the other flits of the packets. All flits in the same packet are transmitted in order as pipelined fashion as shown in Fig. 1. Only the header flit knows where the packet is going, and the remaining data flits must follow the header flit. In wormhole routing, when the needed channels are busy by another broadcast and the flits blocked, then the message will be blocked and can't be delivered to their destination nodes. Deadlock is occurs when a set of messages is blocked forever in the network because each message in the set holds one or more resources needed by another message in this set. No communication can occur over the deadlocked channels until exceptional action is taken to break the deadlock [12] and [13]. Based on these features, many several routing algorithms have been proposed for wormhole communications networks [14], [15], [16], [17], [18], [19], [20], [21] [22], [23], [24] and [25].

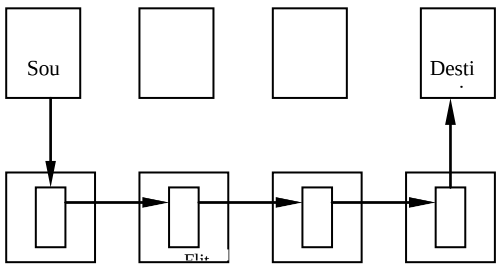


Fig. 1 Wormhole Routing

The main facility of the wormhole routing is the communication latency, where it independent of the distance between the source and destination nodes; 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 [26].

Multicomputer generation has an essential pattern which is called the broadcast routing. Broadcast is compatible to one-to-all communication in which the source send the same message to all possible destination nodes in the network. Efficient broadcast communication is widely used. This is because it's useful in message-passing applications, and is also necessary in several other operations, such as replication and barrier synchronization [27], which are supported in data parallel languages. Broadcast algorithms in wormhole-routed networks can be easily founded in the following researches [28], [29] [30] and [31]. 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 [32].

This paper uses a 3-D mesh topology with Bi-directional channels. A graph M (V, E) can be used to model a 3-D mesh topology where processor can be represented by each node in V (M) and communication channel can be represented by each edge in E (M). The mesh graph is formally defined below. Where m (rows) x n (columns) x r (layers) 3-D mesh comprises mnr nodes connected in a grid fashion

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

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

In the mesh topology the nodes are not necessary connected to the same number of neighbors; each node on the network connects to at least two other nodes. Those at the corners, edges, and middle of the network have four and six neighbors respectively; each node in the mesh consists of its processing element (PE) and its router. The processing element contains a processor and some local memory. There are local channels used by the processing element to inject/eject messages to/from the network, respectively. The injection channel inject the messages that generated by the PE into the network. Three kinds of ports can be distinguished in mesh topology. One-Port, Multi-Ports and All-Ports. This study use the Multi-Port router model where routers can send/receive multiple messages simultaneously to a neighboring node, and that node can simultaneously send/receive messages along all ejection and injection channels [33].

Broadcast problem can be applied by using a Hamilton path from source to destinations. In this paper we introduce the new broadcast algorithm based on the Hamiltonian model for 3-D mesh multicomputers. The rest of the paper is organized as follows. Section 2 presents the related works. Section 3 presents the Hamiltonian model to the 3-D mesh networks and in this Section we introduce the new broadcast algorithm based on the Hamiltonian model while Section 4 examines the performance of the proposed algorithms. Finally, Section 5 concludes this study.

II. RELATED WORKS

The basic function of an interconnection network is routing algorithm which determines the path from a source node to a destination node in a particular network [34]. The dynamic communication performance of a specific network is dependent on a routing algorithm. By developing a good routing algorithm, both throughput and latency can be improved.

Many properties of the interconnection network are a direct consequence of the routing algorithm used. Among these properties the most important ones are as follows:

Connectivity: The system can be able to send and receive packets (messages) from a source node to any number of destination nodes.

- Deadlock and livelock freedom: The ability of the system to guarantee that packets will not block or wander through the network forever.
- Adaptivity: Ability of the system to send packets through alternative paths in the networks.
- Fault tolerance: The system can be able to route packets in the network even if there is any of faulty components.

A. Dual-Path Algorithm

Hamiltonian paths are fundamental of network partitioning strategy based on the deadlock-free routing algorithms [35]. In this algorithm, each node u in a multicomputer is assigned a label, L (u). For each node in the networks, Dual-path assigns it a label based on the position of that node in a Hamiltonian path. For each node in an m x n 2-D mesh the label assignment function L could be described in terms of the x-, y- coordinates as follows:

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

In dual-path algorithm, the network is divided into two subnetworks N_U and N_L by the source, where N_U contains all channels where their routing directions are from the nodes labeled from low to high number and N_L contains all channels where their routing directions are from the nodes labeled from high to low numbers. The routing schema that used to transfer the message in dual-path is made according to the equation that presented in [35]. The routing function can be defined for a source node u and destination node v as R (u, v) = w, such that w is a neighboring node of u, and, if L (u) < L (v), then we have the following equation:

$$L(w)=\max\left\{L(z)\,\middle|\,L(z)\le L(v)\ and\ z\ is\ a\ neighboring\ node\ of\ u\right\}$$
$$or, if\ L(u)>L(v),\ then\ we\ have\ the\ following\ equation:$$
$$L(w)=\min\left\{L(z)\,\middle|\,L(z)\ge L(v)\ and\ z\ is\ a\ neighboring\ node\ of\ u\right\}$$

In dual-path algorithm, the 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 source sends the message to destination nodes in D_U using channels in N_U subnetwork and to destination nodes in D_L using channels in N_L subnetwork.

The destination nodes in D_U is sorted in ascending order by the source and the destination nodes in D_L is sorted in descending order by the source, where the source using the L value as the key in both cases. Two messages were constructed and sent into tow disjoint subnetworks N_U and N_L, one containing D_U as part of the header and the other containing D_L as part of the header..

To make a decision at each node in the mesh, the dual-path routing algorithm uses a distributed routing function R that presented in [35]. When intermediate node receives the message, it determines whether its address matches that of the first destination node in the message header. If the address is match then this address is removed from the message header and the message is copied and sent together with its header to the above (below) neighboring using the routing function R. If the sets of the destination nodes are not empty, the algorithm continues according to the previous method.

B. 3-DHB Algorithm.

The 3-DHB algorithm exploits the features of Hamiltonian paths (using Humiliation label equation for 3-D mesh that is presented in [36] to implement broadcast in two message-passing steps. The source in 3-DHB algorithm divide the network into two subnetworks N_U and N_L, in subnetwork N_U the routing direction is from nodes that their labeled from low to high number and in N_L the routing direction is from nodes that their labeled from high to low numbers [36]. In 3-DHB the destination set D is divided by the source node into two subsets, D_U and D_L. All destination nodes in D_U are located in N_U and all destination nodes in D_L are located in N_L. The message will be sent to destination nodes in D_U by the source using channels in N_U subnetwork and to destination nodes in D_L using channels in N_L subnetwork.

The source sort the destination nodes in D_U in ascending order and the destination nodes in D_L in descending order, where L value is the key label in both cases. The source construct the two messages and sent them into tow disjoint subnetworks N_U and N_L, one containing D_U as part of the header and the other containing D_L as part of the header [36].

C. 3-DSPHB Algorithm

The 3-DSPHB algorithm for All-Port 3-D mesh based on the Hamiltonian paths to implement broadcast in six message-passing steps. The major concept of All-Port architecture is that the local processor can send and receive multiple messages simultaneously. The difference between 3-DHB and 3-DSPHB routing algorithm is that how many message preparation can be operated at the source node. In the 3-DSPHB routing algorithm the destination sets D_U and D_L of the 3-DHB algorithm are further partitioned. The source divide the set D_U into six subsets, The first one containing the all nodes whose x coordinates are greater than or equal the x coordinate of a source, the second one containing the all nodes whose x coordinates are smaller than the x coordinate of a source, the third one containing the all nodes whose y coordinates are greater than or equal the y coordinate of a source, the fourth one containing the all nodes whose y coordinates are smaller than the y coordinate of a source, the fifth one containing the all nodes whose z coordinates are greater than or equal the z coordinate of a source and the sixth one containing the all nodes whose z coordinates are smaller the z coordinate of a source. The source divides D_L as the same partitioned schema that expressed for D_U [36].

III. THE PROPOSED WORKS

A. HAMILTONIAN Models to the 3-D Mesh Networks

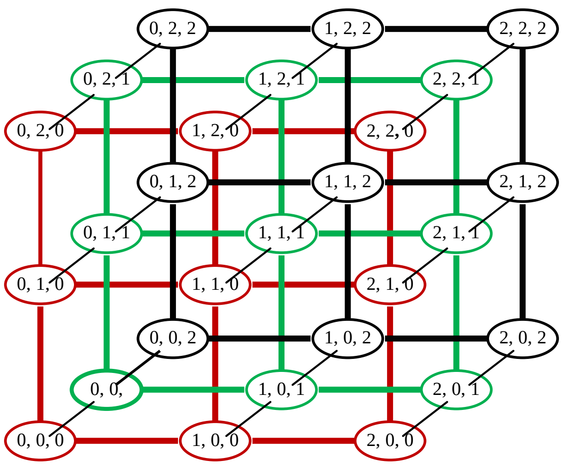
Hamiltonian-path routing is considered as one of the most efficient important approach for designing a deadlock-free algorithm of wormhole-routed network. So the routing algorithm assumed in this paper is the Hamiltonian-path routing. In this routing algorithm, at first, a Hamiltonian path is embedded onto target 3-D mesh network topology, and then messages are routed on the basis of the Hamiltonian path.

The 3-D mesh network can be divides into set of 2-D layers. If we have m x n x r 3-D mesh, then we have r layers of m x n 2-D mesh where each layer will be labeled with Hamilton model for 2-D mesh as shown in [35]. The label assignment function ℓ for an m x n mesh can be expressed in terms of the x and y coordinates of nodes as

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

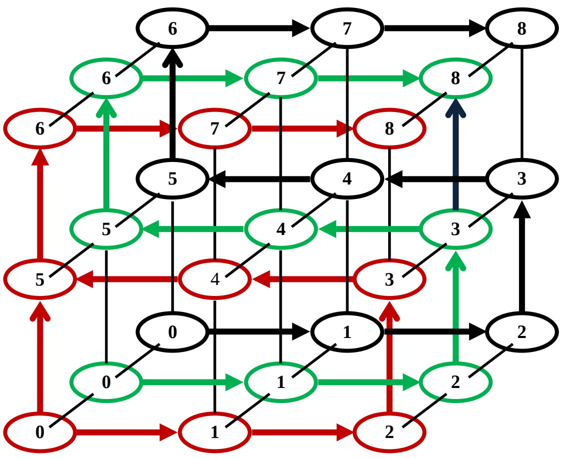
The network partitioning strategy is fundamental to our broadcast routing algorithms. Using the Hamiltonian model labeling, the 2-D network can be partitioned into two subnetworks. The first one is called high-channel network and contains all channels where their routing directions are from the nodes labeled from low to high number. The second one is called the low-channel network and contains all channels where their routing directions are from the nodes labeled from high to low numbers. According to this partition schema, each 2-D network will be divided into two disjoint subnetworks and each subnetwork has its independent set of physical links in the network.

A Hamiltonian labeling strategy for 3 x 3 x 3 3-D mesh can be shown in Fig. 2(a), where every node can be represented by its integer coordinate (x, y, z). A high-channel subnetwork which contains all channels whose direction is from lower-labeled nodes to higher-labeled nodes can be shown in Fig. 2(b). A low-channel subnetwork which contains all channels whose direction is from higher-labeled nodes to lower-labeled nodes can be shown in Fig. 2(c)



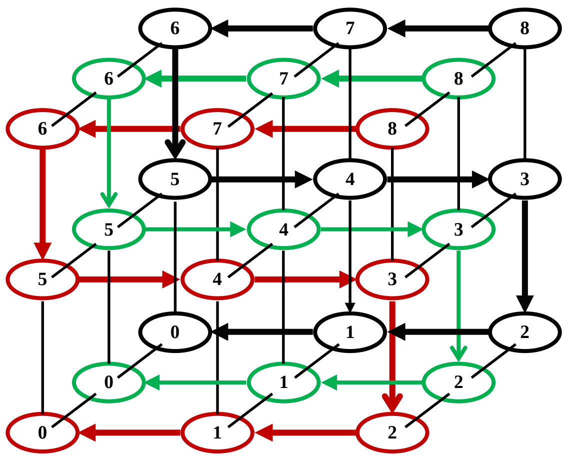
(a)

Fig. 2 The labeling of a 3 x 3 x 3 mesh. (a) Physical network



(b)

Fig. 2 The labeling of a 3 x 3 x 3 mesh. (b) High-Channel network



(c)

Fig. 2 The labeling of a 3 x 3 x 3 mesh. (c) Low-Channel network

B. *The Proposed Algorithm*

This section introduces a solution for the broadcast wormhole routed mesh multicomputer. The 3-DTPMP algorithm based on the Hamiltonian paths to implement broadcast in Multi-Port. A major consequence of Multi-Port architecture is that the local processor may transmit (receive) multiple messages simultaneously.

Our proposed 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 presented in Section 3.1. 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:

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

Suppose that the coordinate of the source node u_0 is represented by (x_0, y_0, z_0) with the Hamiltonian labeling $\ell(u_0)$, and D represents the all nodes in the 3-D mesh with the Hamiltonian labeling ℓ .The simple idea of this algorithm is as follow: -

Step 1: The source split the D into two subsets D_{Z0} and D_{ZR} , where D_{Z0} contain the nodes which their z coordinates are equal z_0 and D_{ZR} contain the reminder nodes which their z coordinates are not equals z_0 . Formally, the subsets are described by the following expression:

$$D_{Z0} = \{(x,y,z)|(x,y,z)\in D,(0 \le x < m),(0 \le y < n),(z = z_0)\},$$
$$D_{ZR} = \{(x,y,z)|(x,y,z)\in D,(0 \le x < m),(0 \le y < n),(z \ne z_0)\},$$

Step 2: The subnet that contains the nodes of D_{Z0} represent a layer of 2-D mesh and will be labeled with Hamilton model that presented in Section 3.1. The source split D_{Z0} into two subsets, D_U and D_L, where D_U contain the nodes which their label is greater than label of z_0, and D_L contain the nodes which their label is smaller than label of z_0. The source sorts 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.

Step 3: The source split the D_U subset into set of surfaces. In fact, if we have m x n x r 3-D mesh, then we have m surfaces of mesh. Formally, the sub-networks are described by the following expression:

$$D_{x0} = \{(x,y,z)|(x,y,z)\in D_{Z0}(x=0),(0 \le y < n),(0 \le z < r)\},$$
$$D_{x1} = \{(x,y,z)|(x,y,z)\in D_{Z0}(x=1),(0 \le y < n),(0 \le z < r)\},$$
$$\vdots$$
$$D_{xm-1} = \{(x,y,z)|(x,y,z)\in D_{Z0}(x=m-1),(0 \le y < n),(0 \le z < r)\}$$

Step 4: The source will sends the message to all nodes in D_{x0} using Hamilton model of 2-D mesh, for all nodes in D_{x1} using Hamilton model of 2-D mesh and so on. The same partition schema is repeated for D_L.

Step 5: The source divide D_{ZR} into subsets, D_{R1}, D_{R2}, ……….. D_{RN} where each subset represents a layer of 2-D mesh and will be labeled with Hamilton model that presented in Section 3.1. Formally, the subsets are described by the following expression:

$$D_{R1} = \{(x,y,z) | (x,y,z) \in D_{ZR}, (0 \le x < m), (0 \le y < n), (0 = z \ne z_0)\},$$
$$D_{R2} = \{(x,y,z) | (x,y,z) \in D_{ZR}, (0 \le x < m), (0 \le y < n), (1 = z \ne z_0)\},$$
.
.
.
$$D_{RN} = \{(x,y,z) | (x,y,z) \in D_{ZR}, (x \le 0 < m), (0 \le y < n), (r-1 = z \ne z_0)\}$$

Step 6: The source split D_{R1} into two subsets, D_{R1U} and D_{R1L}, where D_{R1U} contain the nodes which their label is greater than label of z_0, and D_{R1L} contain the nodes which their label is smaller than label of z_0. The source sort the destination nodes in D_{R1U}, using the L value as the key, in ascending order. Sort the destination nodes in D_{R1L}, using the L value as the key, in descending order

Step 7: The source split the D_{R1U} subset into set of surfaces using the same schema that presented in step 3.

Step 8: Each above subset of D_{R1U} represents a surface. The source (x_0, y_0, z_0) sends the message to a nodes, which are represented by their integer coordinates (x_0, y_0, z, where $z \neq z_0$) which will be act as a source on its surface. Each alternative source will send the message to all nodes in its surface (x, y_0, z, where x equals value from 0 to m-1) using Hamilton model of 2-D mesh

Step 9: The source divide D_{R1L} into subsets as the same schema that used for D_{R1U}, and repeat step 7 and step 8 for each subset in D_{R1L}.

Step 10: Repeat the steps from 5 to step 9 for the remaining subsets D_{R2}, D_{R3},….. and D_{RN}.

Theorem 1: 3-DTPMP 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-DTPMP algorithm is deadlock-free at r subnetworks. Now, we will prove that there are no dependencies within each subnetwork. The 3-DTPMP algorithm uses Hamiltonian-path routing to distribute destination nodes for each subnetwork, as proved in [36] Hamiltonian-path algorithm is deadlock-free; therefore, no cyclic dependency can exist among the channels. Hence 3-DBL algorithm is deadlock-free.

C. Comparative Study

To demonstrate the performance of 3-DTPMP 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

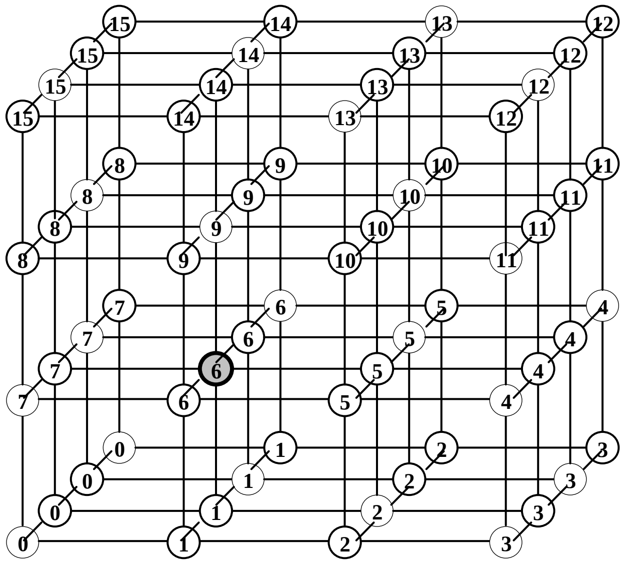


Fig. 3 An example of 4x4x4 mesh

The 3-DTPMP algorithm splits D into two subsets D_{Z0}, D_{ZR} 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_{ZR} = { (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), (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 routing pattern of 3-DTPMP algorithm is shown with bold lines in Fig. 4(a) for 2-D mesh where z coordinate =0, in Fig. 4(b) for 2-D mesh where z coordinate =1, in Fig. 4(c) for 2-D mesh where z coordinate =2 and in Fig. 4(d) for 2-D mesh where z coordinate =3.

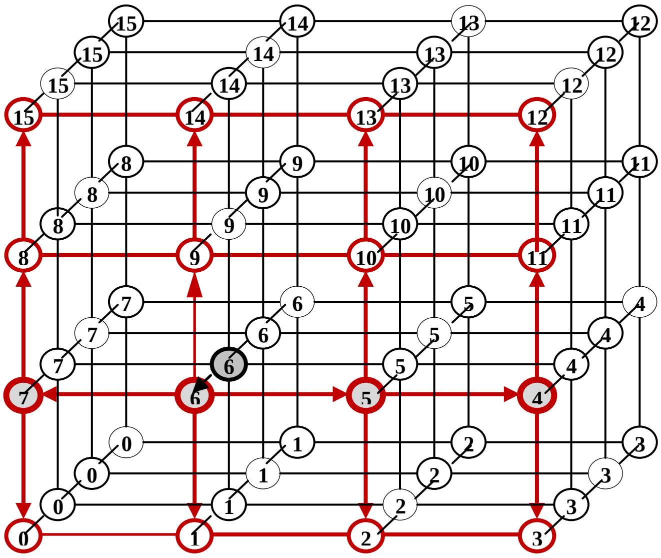


Fig. 4 (a). The routing patterns of 3-DTPMP algorithm, for 2-D mesh where z coordinate = 0

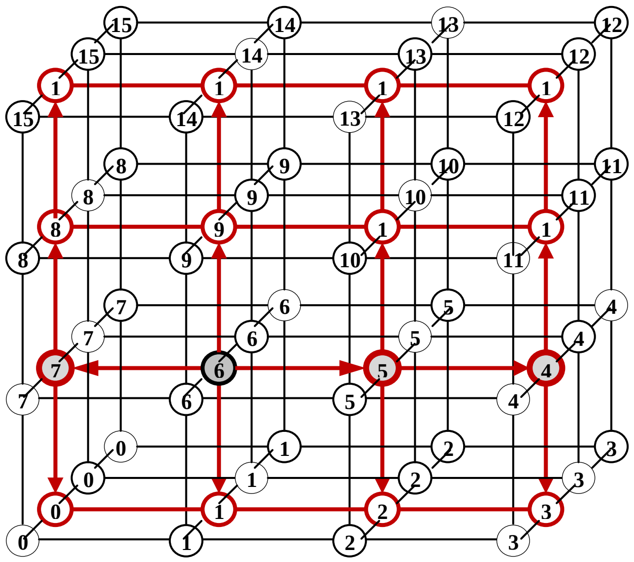


Fig. 4 (b). The routing patterns of 3-DTPMP algorithm, for 2-D mesh where z coordinate = 1

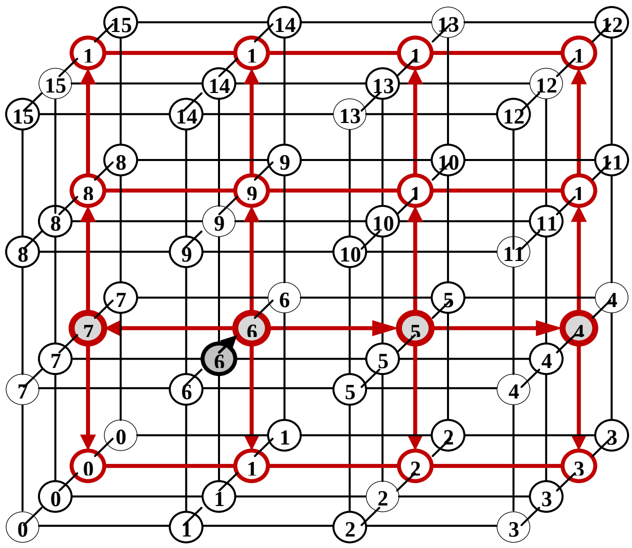


Fig. 4 (c). The routing patterns of 3-DTPMP algorithm, for 2-D mesh where z coordinate = 2

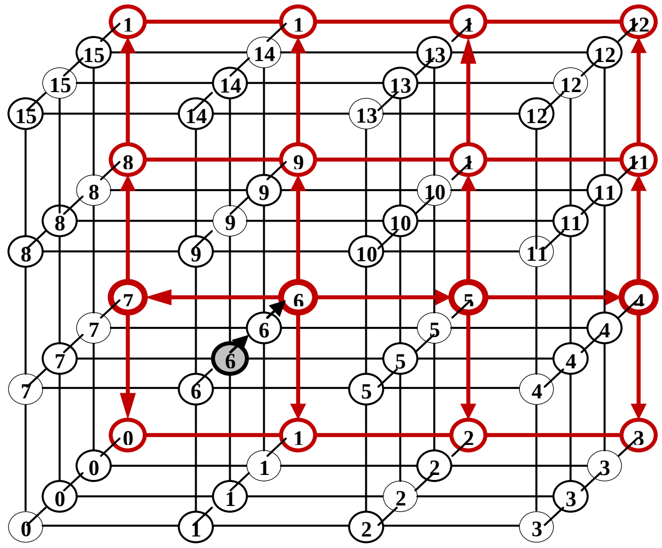


Fig. 4 (d). The routing patterns of 3-DTPMP algorithm, for 2-D mesh where z coordinate = 3

IV. SIMULATIONS

In order to compare the performance of our proposed broadcast routing algorithm, 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 [37]. CSIM is used a multiple processes to execute testing in parallel fashion and provides a very realistic environment which can be used in modular simulation programs. The simulation program for broadcast communication is part of a larger simulator, called MultiSim [38]. MultiSim was used to simulate broadcast operations in 3D mesh of different sizes.

The performance of our proposed algorithm with broadcast routing is studied by using large number of different source nodes randomly and different size of messages with different injection rate. All simulations were performed for a 6 x 6 x 6 3-D mesh. The software that used to buffers allocating, messages, copy message and router initializing, etc overhead is called the startup latency ß. The message is divided into flits, so the number of these flits in a message denotes the message length.

In order to evaluate our proposed algorithm 3-DTPMP, we compare its performance with 3-DHB and 3-DSPHB algorithms that is presented in [36]. The average broadcast latencies of three algorithms across different message lengths (100 bytes to 2000 bytes) and startup latency ß is set to 100 microseconds are shown in Fig. 5. The advantage of the 3-DTPMP algorithm is significant. In 3-DTPMP the source node divided the net into r subnets (layers of 2-D networks) and each layer of 2-D net works is divided into set of surfaces (m surfaces) as shown in Section 3.1, so the path from source node to destination nodes in each surface is less than the path in the 3-DHB and 3-DSPHB routing algorithms; this advantage can be shown in termed of generated traffic. Since communication latency in wormhole-routed broadcast systems is dependent on the length of the message, average latencies of 3-DTPMP algorithm is less sensitive as the length of the message as shown in Figure 5. A good behavior of the algorithm path is depends strongly on the start-up time that is required for each algorithm. Because 3-DSPHB is All-ports it takes six startup latency times while 3-DHB takes only two start up latency times, so The 3-DHB is peter performance than 3-DSPHB and the length of the path to reach the destination is decreasing in 3-DHB algorithm.

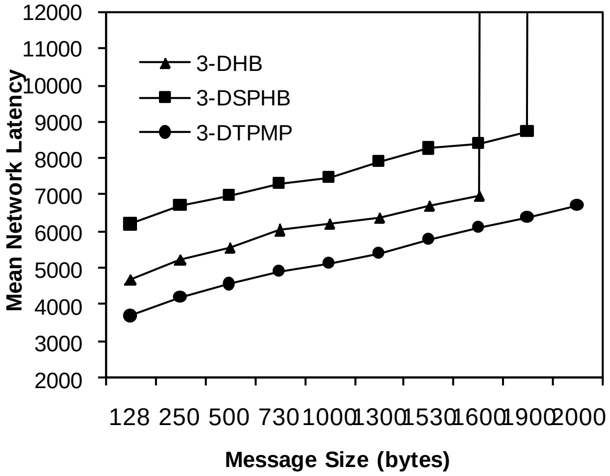


Fig. 5 Performance Comparison of 3-DTPMP algorithm with 3-DHB and 3-DSPHB algorithms under different message size

Figs 6 shows the network latency that obtained by the three algorithms where we used various network loads. We have fixed the startup latency ß as 100 microseconds and we have fixed the message length as 300 flits. The performance of our proposed algorithm 3-DTPMP is the best. Fig. 6 shows us that the disadvantage behavior of the 3-DSPHB algorithm is increases with the injection rate increased.

The source node in 3-DTPMP divide the destination nodes into 4 subset, in 3-DHB the source divide the destination nodes into two subsets while in 3-DSPHB divide destination nodes into six subsets, so when any flits from another broadcast message is send and the recent broadcast transmission is not complete, then these flits will be blocked until the recent broadcast is complete. In fact, if the load is very high, the 3-DSPHB may decrease system throughput and increase message latency. Because the 3-DTPMP are shortest paths from source to destinations (set of individual layers), its performance dose not effect when loads on net is very high. So 3-DTPMP is the best good performance. In the 3-DHB when the broadcast size increases, the number of message-passing steps required to complete the broadcast operation is not increase, so 3-DHB algorithm is significant. The source node in 3-DHB algorithm will send two messages only from its two external channels. Since communication latency in wormhole-routed broadcast systems is dependent on the injection rate, average latencies of 3-DSPHB algorithm linearly increase as the length of the message as shown in Figure 6.

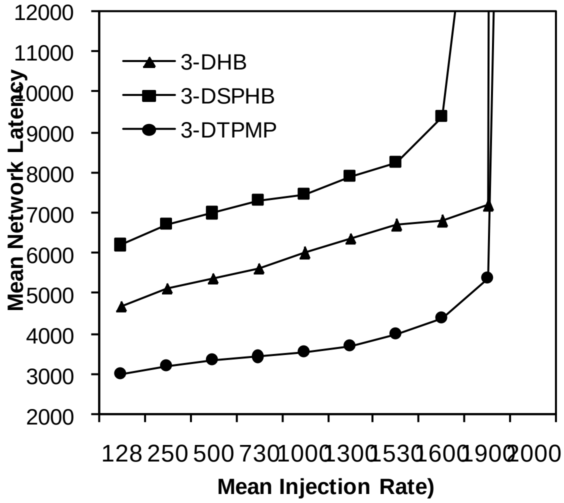


Fig. 6 Performance Comparison of 3-DTPMP algorithm with 3-DHB and 3-DSPHB algorithms under different loads

V. CONCLUSION

Broadcast is one of the most important communication operation that can be used in multicomputer system and can be applied for different algorithms such as parallel drawing algorithms, fast Fourier switch, and Fence synchronization method.

In this paper, we propose a new algorithm for broadcasting messages on Multi-Port wormhole-routed 3-D mesh with arbitrary size. This algorithm is shown to be deadlock-free. The performance study shows that the 3-DTPMP algorithm outperforms than 3-DHB and 3-DSPHB algorithms for different length of messages and different injection rate with reduced delays. Our proposed algorithm is better than the presented algorithms, because the paths from source to destinations are shortest. Our proposed algorithm can be developing and extending to higher dimensional networks.

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/cray t3e/.

[9] W. Dally and C. Seitz, (1986) The Torus Routing Chip, J. Distributed Computing, Vol. 1, No. 3, pp 187-196.

[10] Ammar Karkar, Nizar Dahir, Ra’ed Al-Dujaily, Kenneth Tong, Terrence Mak, and Alex Yakovlev, (2014) Hybrid Wire-Surface Wave Architecture for One-to-Many Communication in Networks-on-Chip, Journal of Parallel and Distributed Computing, Volume 61, Pages 1307-1336

[11] William James Dally, Brian Towles, (2004) Principles and Practices of Interconnection Networks”, Morgan Kaufmann Publishers, 13.2.1 "Inc. ISBN 978-0-12-200751-4.

[12] 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.

[13] 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.

[14] H. Moharam, M. A. Abd El-Baky & S. M. M., (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.

[15] Nen-Chung Wang, Chih-Ping Chu & Tzung-Shi Chen, (2002) A dual hamiltonian-path-based multicasting strategy for wormhole routed star graph interconnection networks, J. Parallel Distrib. Comput. Vol. 62, pp 1747–1762.

[16] 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.

[17] E. Fleury, P. Fraigniaud, (1998) Strategies for path-based multicasting in wormhole-routed meshes, Journal of Parallel & Distributed Computing, Vol. 6, pp 26–62.

[18] P. McKinley, Y. J. Tsai. D. Robinson, (1995) Collective communication in wormhole-routed massively parallel computers, IEEE Computer, Vol. 28, No. 12, pp 39–50.

[19] Rakesh Matam and Somanath Tripathy, (July 2013) WRSR: wormhole-resistant secure routing for wireless mesh networks, EURASIP Journal on Wireless Communications and Networking, DOI: 10.1186/1687-1499-2013-180

[20] Jonny Karlsson, Laurence S. Dooley and Göran Pulkkis, (May 2013) Identifying Time Measurement Tampering in the Traversal Time and Hop Count Analysis (TTHCA) Wormhole Detection Algorithm Sensors (Basel), Published online 2013 May 17. doi: 10.3390/s130506651.

[21] Dianne R. Kumar , Walid A. Najjar & Pradip K. Srimani, (2001) A New Adaptive Hardware Tree-Based Multicast Routing in K-Ary N-Cubes, IEEE Transactions on Computers, Vol.50 No.7, pp 647-659.

[22] Jianxi Fan, (2002) Hamilton-connectivity and cycle-embedding of the Mobius cubes, Information Processing Letters, Vol. 82 No. 2, pp113-117.

[23] Amnah El-Obaid and Wan-Li Zuo (2007), Hamiltonian Paths for Designing Deadlock-Free Multicasting Wormhole-Routing Algorithms in 3-D Meshes, Journal of Applied Sciences, 3410-3419. DOI: 10.3923/jas.2007.3410.3419

[24] Jung-hyun Seo and HyeongOk Lee, (17 November 2013) Link-Disjoint Broadcasting Algorithm in Wormhole-Routed 3D Petersen-Torus Networks, International Journal of Distributed Sensor Networks, Volume 2013 (2013), Article ID 501974, 7 pages

[25] Patricia Ruiz,(September 2015) Survey on Broadcast Algorithms for Mobile Ad Hoc Networks,
ACM Computing Surveys (CSUR), Volume 48, Article No. 8

[26] 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.

[27] Tim S. Axelrod, (1986) Effects of synchronization barriers on multiprocessor performance, Parallel Computing, Vol. 3, No. 2, pp 129-140.

[28] 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.

[29] Yuh-Shyan Chen & 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.

[30] 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**.**

[31] Z. Shen, (2007) " A generalized broadcasting schema for the mesh structures", Applied Mathematics and Computation, Vol. 186, No. 2, pp 1293–1310.

[32] Z. Shen, (2007) A generalized broadcasting schema for the mesh structures, Applied Mathematics and Computation, Vol. 186, No. 2, pp 1293–1310.

[33] 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.

[34] 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.

[35] Lin X, L. Ni M, (1991) "Deadlock-free multicast wormhole routing in multicomputer networks" ,In 18th Annual International Symposium on Computer Architecture, Toronto, pp 116-125

[36] Amnah El-Obaid, (2015), Three-Dimension Hamiltonian Broadcast Wormhole-Routing, International Journal of Computer Networks & Communications (IJCNC), Vol.7, No.3.

[37] H. D. Schwetman, (1985) "CSIM: A C-based, process-oriented simulation language, Tech. Rep." Microelectronics and Computer Technology Corp, PP 80-85.

[38] P. K. McKinley & C. Trefftz, (1993) "MultiSim: A tool for the study of large-scale multiprocessors, in Proc", Int. Workshop on Modeling, Analysis and Simulation of Comput. and Telecommun. Nehvorks (MASCOTS 93), pp 57-62.

Authors: Amnah Elobaid **r**eceived 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.