

THREE-DIMENSION HAMILTONIAN BROADCAST WORMHOLE-ROUTING

Amnah El-Obaid

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

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. Broadcasting is a data communication task in 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. In this paper, we propose an efficient two algorithms for broadcasting on an all-port wormhole-routed 3D mesh with arbitrary size. 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 introduce an efficient two algorithms, Three-Dimension Hamiltonian Broadcast (3-DHB) and Three-Dimension Six Ports Hamiltonian Broadcast (3-DSPHB) which used broadcast communication facility with deadlock-free wormhole routing in general three-dimensional networks. In this paper the behaviors of these algorithms were compared using simulation. The results presented in this paper indicate that the advantage of the proposed algorithms.

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 where it is used efficiently in weather forecast and high performance servers. 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. Some sample topologies are shown in Fig. 1. 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].

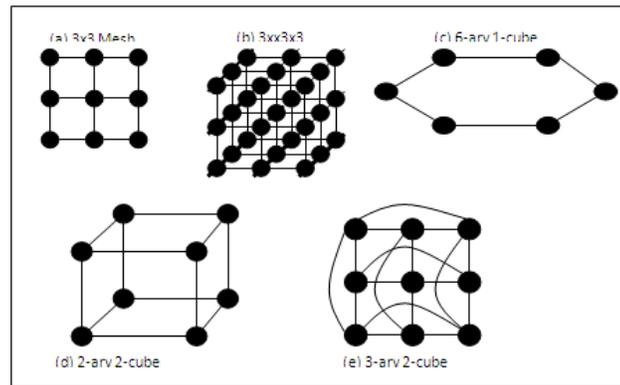


Fig. 1 Multicomputer network topologies.

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. 2. 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], [13]. Based on these features, many several routing algorithms have been proposed for wormhole communications networks [14], [15], [16], [17], [18], [19], [20], [21] and [22].

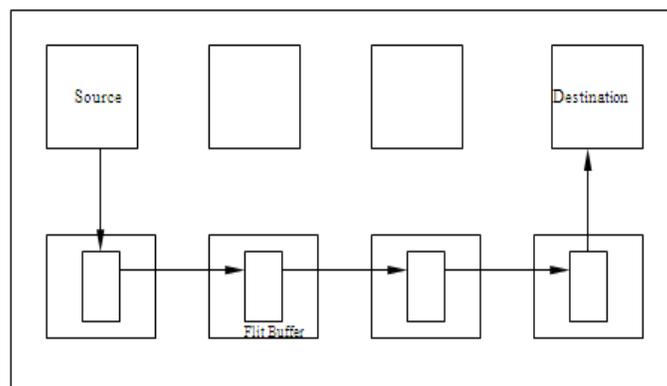


Fig. 2 Wormhole Routing

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 [23].

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 [24], which are supported in data parallel languages. Broadcast algorithms in wormhole-routed networks can be easily founded in the following researches [25], [26] [27] and [28]. 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 [29].

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) \times n (columns) \times r (layers) 3-D mesh comprises mnr nodes connected in a grid fashion

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

$$V(M) = \{ (x, y, z) \mid 0 \leq x < n, 0 \leq y < m, 0 \leq z < r \} \text{ and}$$

$$E(M) = \{ [(x_i, y_i, z_i), (x_j, y_j, z_j)] \mid (x_i, y_i, z_i), (x_j, y_j, z_j) \in V(G),$$

$$\text{and } |x_i - x_j| + |y_i - y_j| + |z_i - z_j| = 1 \}$$

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 All-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 [30].

Broadcast problem can be applied by using a Hamilton path from source to destinations. In this paper we introduce the new broadcast algorithms 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. In Section 4 we

introduce the new broadcast algorithms based on the Hamiltonian model while Section 5 examines the performance of the proposed algorithms. Finally, Section 6 concludes this study.

2. 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 [31] and [32]. 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.

2.1. The XY routing algorithm

In a 2D mesh, each node is represented by its position (x, y) in the mesh [33]. The packets in the XY routing algorithm will be sent in the X dimension first and then in the Y dimension by the source. Let (s_x, s_y) represent the addresses of a source node and (d_x, d_y) represent the addresses of destination node. Furthermore, let $(g_x, g_y) = (d_x - s_x, d_y - s_y)$. The XY routing algorithm place g_x , and g_y in the first two flits, respectively, of the packet. When the first flit of a packet arrives at a router, it is decremented or incremented, depending on whether it is greater than 0 or less than 0. The source sends the packet forward in the same direction when the result is not equal to 0. If the result equals 0 and the packet arrived on the Y dimension, the packet is delivered to the local processor. If the result equals 0 and the packet arrived on the X dimension, the flit is discarded and the next flit is examined upon arrival. When that flit is 0 the local processor receives the packet; otherwise, the packet routing in Y dimension. Using this method, the largest possible 2D mesh with an 8-bit flit is 128 x128.

2.2. Dual-path algorithm

Hamiltonian paths are fundamental of network partitioning strategy based on the deadlock-free routing algorithms [34]. 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 $m \times n$ 2-D mesh the label assignment function L could be described in terms of the x -, y - coordinates as follows:

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

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 [34]. 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:

$$L(w) = \max \{L(z) \mid L(z) \leq L(v) \text{ and } z \text{ is a neighboring node of } u\}$$

or, if $L(u) > L(v)$, then we have the following equation :

$$L(w) = \min \{L(z) \mid L(z) \geq L(v) \text{ and } z \text{ is a neighboring node of } u\}$$

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 [34]. 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.

2.3. Multi-path algorithm

The Multi-Path algorithm is all-port, where it sends the message by using the 4 outgoing degree channels. The difference between multi-path and dual-path routing algorithm is that how many message preparation can be operated at the source node. In the multipath routing algorithm the destination sets D_U and D_L of the dual-path algorithm are further partitioned. The source divide the set D_U into two subsets, The first one containing the all nodes whose x coordinates are greater than or equal the x coordinate of a source and the second one containing the all nodes whose x coordinates are smaller than the x coordinate of a source. The same partitioned schema can be done for set D_L .

In Multi-path routing algorithm the path from source node to destination nodes is less than the path in the Dual-path routing algorithm; this advantage can be shown in termed of generated traffic. Also, the numbers of channels that use by a Multi-path routing algorithm are fewer than number of channels that use by dual-path routing algorithm. A detailed performance study can be found in [34] and [35].

2.4. The Sequential Multi-Column algorithm (SMC)

The basic idea of this algorithm is by splitting the destination nodes into groups of columns and send the message to these different groups in a pipeline fashion. The term “sequential” refers to the use of only one port at a time between the two ports available to reach higher labels. An instance of the split-column-function for an integer p is a set of p groups of consecutive columns S_1, S_2, \dots, S_p , labeled from 1 to p from left to right, with $S_i \cap S_j = \emptyset$, and such that the cardinality (that is the number of columns) of the different groups differs by at most one.

At step 1, The source U_0 send the message to destination nodes belonging to S_1 ; then, when the last flit of the message leaves the source, send (at step 2) the message to destination nodes belonging to S_2 ; and so on: at step c , send the message to the destination nodes belonging to the group of columns S_c . Step c begins at time $(c-1)(\beta+L)$. Each copy of the message needs $\beta + L$ to leave the source node, where β represent the ratio of the startup time over the propagation time of a flit from one router to a neighboring router. If σ were any permutation of the subsets (S_1, S_2, \dots, S_p) , then (at step 1) send the message to destination nodes belonging to $S_{\sigma^{-1}(1)}$. Then (at step 2), the source node will send the message to destination nodes when the last flit of the message leaves it, and so on. At step t , send the message to the destination nodes belonging to the group of columns $S_{\sigma^{-1}(t)}$. In other words, the message to group c is sent at step $\sigma(c)$. The performance of the SMC algorithm depends on the number of group's p , and on the schedule determined by the permutation σ for sending the message to these groups [35].

3. 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.

A 3-D mesh network contains many Hamiltonian paths. In the following, we give the node labeling function $\ell(u)$ for a 3-D mesh. For an $m \times n \times r$ 3-D mesh we can assign a label function ℓ to each node in 3-D mesh which can be expressed in terms of the x -, y - and z -coordinates as follows:

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

Each node in 3-D mesh can be assign a unique number using the above labeling function. The Hamiltonian model can be assigned each node in 3-D mesh a unique label starts from 0, followed by 1, and so on until the last node labeled $mnr-1$.

The network partitioning strategy is fundamental to our broadcast routing algorithms. Using the Hamiltonian model labeling, the network can be partitioned into two subnetworks. The first one is

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 .

4.1. Three-Dimension Hamiltonian Broadcast (3-DHB) algorithm

Step 1: The source node divides D into two subsets D_U and D_L as follows:

$$D_U = \{ (x, y, z) \mid (x, y, z) \in D, \ell(x, y, z) > \ell((x_0, y_0, z_0)) \}$$

$$D_L = \{ (x, y, z) \mid (x, y, z) \in D, \ell(x, y, z) < \ell((x_0, y_0, z_0)) \}$$

Step 2: If D_U contains the following nodes $\{d1, d2, d3, \dots, dn\}$, and D_L contains the following nodes $\{d1, d2, d3, \dots, dn\}$, then sorts them according to labels as follows:

$$D_U = \{ \ell(d1) < \ell(d2) < \ell(d3) < \dots < \ell(dn) \}$$

$$D_L = \{ \ell(d1) > \ell(d2) > \ell(d3) > \dots > \ell(dn) \}$$

Step 3: Construct two messages, one containing nodes in D_U as part of the header and the other containing nodes in D_L as part of the header. The source sends the message to subset D_U through subnetwork N_U and sends the message to subset D_L through subnetwork N_L .

Theorem 1: 3-DHB is deadlock-free.

Proof: At the source node, 3-DHB algorithm divides the network into two disjoint subnetworks. This is obvious since, $N_U \cap N_L = \emptyset$. Then 3-DHB algorithm is deadlock-free at the two subnetworks. To complete our proof, we should prove that there are no dependencies in each subnetwork N_U and N_L . In N_U a message entering a node and always leaves on a node with label greater than label of entered node, while in N_L a message entering a node and always leaves on a node with label lower than label of entered node, therefore, no cyclic dependency can exist among the channels. So 3-DHB is deadlock-free.

4.2. Three-Dimension Six Ports Hamiltonian Broadcast (3-DSPHB) algorithm

Step 1: The split schema is the same schema that is presented in the first step of 3-DHB algorithm. Partition D into D_U and D_L

Step 2: Similarly, sorts D_U and D_L as done in the second stage of 3-DHB.

Step 3: Divide D_U into three subsets, D_{U1} , D_{U2} and D_{U3} as follows:

$$D_{U1} = \{ (x, y, z) \mid (x, y, z) \in D_U, x > x_0, (0 \leq y < m), (0 \leq z < r) \},$$

$$D_{U2} = \{ (x, y, z) \mid (x, y, z) \in D_U, x < x_0, (0 \leq y < m), (0 \leq z < r) \},$$

$$D_{U3} = \{ (x, y, z) \mid (x, y, z) \in D_U, x = x_0, (0 \leq y < m), (0 \leq z < r) \}$$

Construct three messages, one containing D_{U1} as part of the header and, second containing D_{U2} as part of the header and the other containing D_{U3} , as part of the header. The source sends three messages to neighboring nodes using the subnetwork N_U .

Step 5: Similarly, partition D_L into D_{L1} , D_{L2} and D_{L3} and construct three messages.

Theorem 2: The 3-DSPHB is deadlock-free.

To prove this theorem, we can use the same schema that is used to prove theorem 1.

4.3. Comparative study

To study and compare the performance of 3-DHB and 3-DSPHB algorithms, consider the example shown in Fig. 4 for a $4 \times 4 \times 4$ mesh topology labeling using a Hamiltonian path. The source node with Hamilton labeled 25 which has an integer coordinate (1, 1, 1) will send a broadcast to all nodes in its 3-D mesh multicomputer.

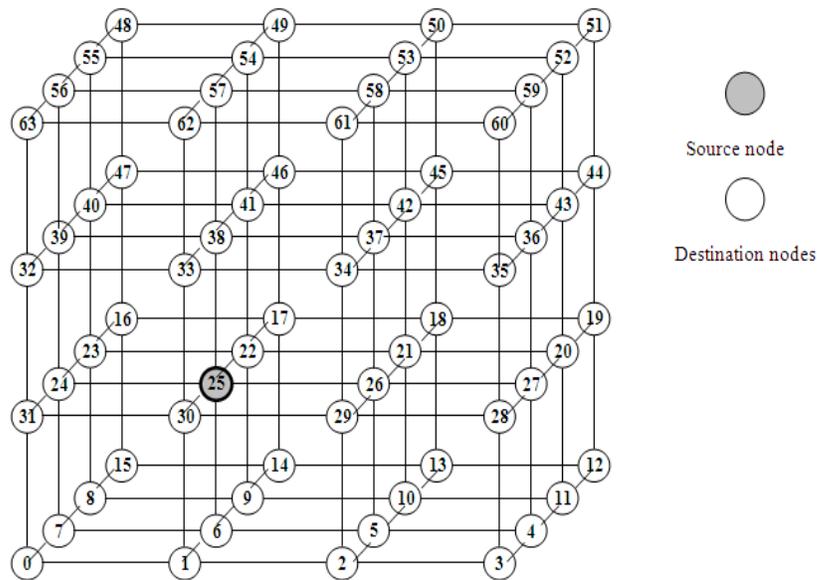


Fig. 4 An example of $4 \times 4 \times 4$ mesh

The source node in 3-DHB algorithm will split and sort D into two subsets D_U and D_L , where every node in D_U has a higher label than the label of the source node and every node in D_L has a lower label than that the label of the source node. The content of D_U and D_L will be as follows:-

$D_U = \{26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63\}$

$D_L = \{24, 23, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0\}$, the

The routing pattern is shown with bold and dashed lines in Fig. 5, where bold lines represent routing in N_U and dashed lines represent routing in N_L .

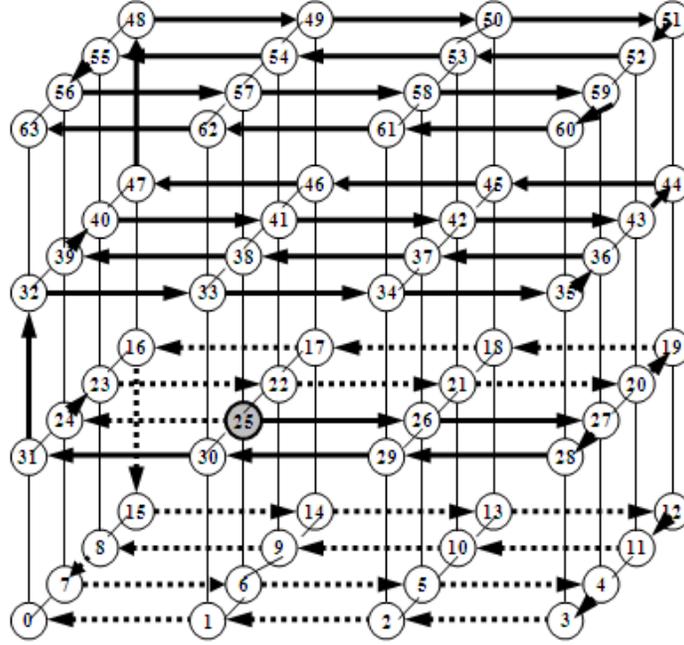


Fig. 5 The routing patterns of 3-DHB algorithm, bold lines of N_U subnetwork, and dashed lines of N_L subnetwork

The 3-DSPHB algorithm splits and sorts D into two subsets D_U where contains every node in N_U that has a higher label than that of the source node and D_L where contains every node in N_L that has a lower label than that of the source node.

$$D_U = \{26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63\}$$

$$D_L = \{24, 23, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0\}$$

D_U is further divided into three subsets D_{U1} , D_{U2} and D_{U3} , with D_{U1} containing the nodes whose x coordinates are lower than the x coordinate of the source, D_{U2} containing the nodes whose x coordinates are greater than the x coordinate of source and D_{U3} containing the remaining nodes in D_U . The routing pattern is shown with bold and dashed lines in Fig. 6.

$$D_{U1} = \{31, 32, 39, 40, 47, 48, 55, 56, 63\}$$

$$D_{U2} = \{26, 27, 28, 29, 34, 35, 36, 37, 42, 43, 44, 45, 50, 51, 52, 53, 58, 59, 60, 61\}$$

$$D_{U3} = \{30, 33, 38, 41, 46, 49, 54, 57, 62\}$$

D_L is further divided into three subsets D_{L1} , D_{L2} and D_{L3} , with D_{L1} containing the nodes whose x coordinates are lower than the x coordinate of the source, D_{L2} containing the nodes whose x

coordinates are greater than the x coordinate of the source and D_{L3} containing the remaining nodes in D_L . The routing pattern is shown with bold and dashed lines in Fig. 7.

$$D_{L1} = \{24, 23, 16, 15, 8, 7, 0\}$$

$$D_{L2} = \{21, 20, 19, 18, 13, 12, 11, 10, 5, 4, 3, 2\}$$

$$D_{L3} = \{22, 17, 14, 9, 6, 1\}$$

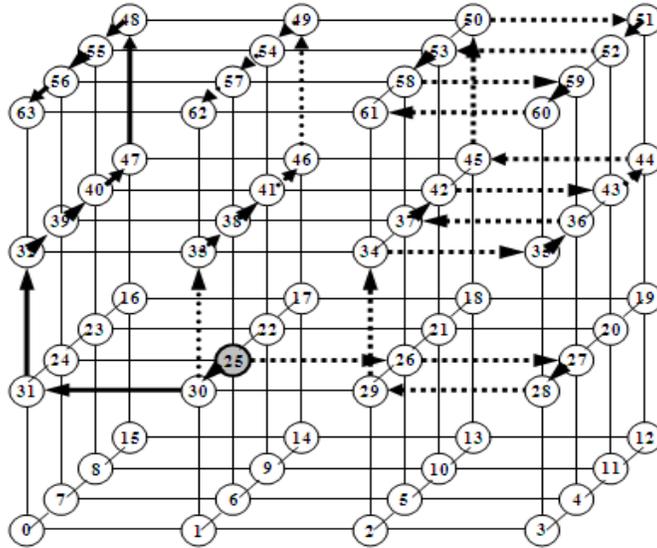


Fig. 6 The Routing Patterns Of 3-Dsph Algorithm Of Nu

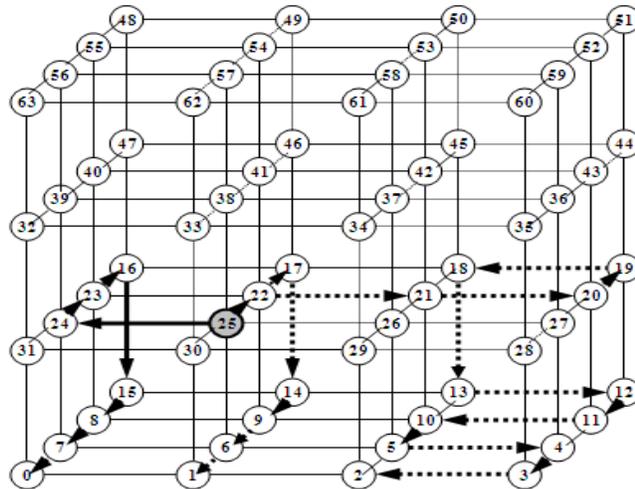


Fig. 7 The routing patterns of 3-DSPHB algorithm of NL

5. SIMULATIONS

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 [36]. 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 [37]. MultiSim was used to simulate broadcast operations in 3D mesh of different sizes. To study the performance of our algorithms with broadcast schema we random a large number of different source nodes with different size of messages and different injection rate. 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 β , message lengths and average injection rate (interarrival time). The message is divided into flits, so the number of these flits in a message denotes the message length and the software that overhead for buffers allocating, messages coping, router initializing, etc. is represented the startup latency. The maximum latencies were measured and then averaged over these samples.

Figure 8 compares the two algorithms across different message lengths (100 bytes to 2000 bytes) in 5 x 5 x 5 network. In Fig 8(a), the startup latency β is set to 10 microseconds. In Fig 8(b), the startup latency β is set to 100 microseconds. In both cases, the advantage of the 3-DHB algorithm is significant. 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 n 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 length of the message, average latencies of 3-DSPHB algorithm linearly increase as the length of the message as shown in Figure 8.

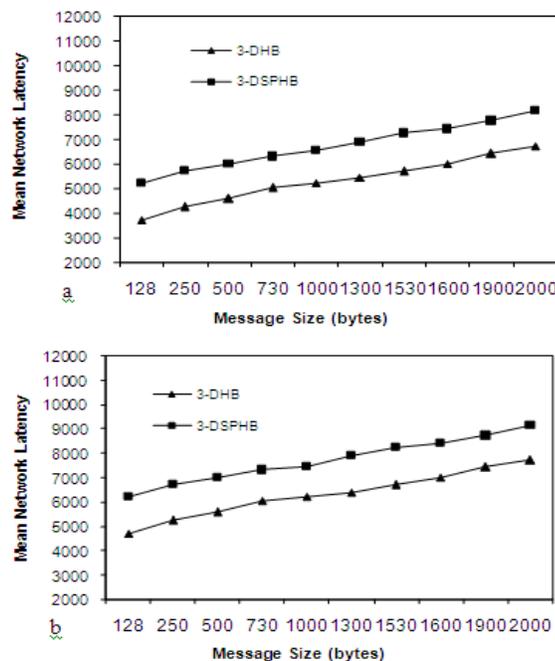


Fig. 8 Comparison of 3-DHB with 3-DSPHB. (a) With small message latency $\beta = 10$; (b) with large message latency $\beta = 100$

For testing the effects of average injection rate (average flow rate time) on our proposed algorithms we present this section experiments. We have fixed the message length as 1000 flits. Fig. 9 compares the two algorithms across different injection rate in 5 x 5 x 5 network. In Fig. 9(a), the startup latency β is set to 10 microseconds. In Fig. 9(b), the startup latency β is set to 100 microseconds.

Two algorithms exhibit good performance at low loads. The effect of our two algorithms 3-DHB and 3-DSPHB is the same and well performance for small start-up time ($\beta = 10$). This is because there is no scramble in the network according to other broadcast. This result can be shown in Fig 9(a).

The disadvantage of 3-DSPHB algorithm increases with the injection rate as shown in Fig 9(b). The 3-DHB algorithm is less sensitive rather than the 3-DSPHB algorithm. In the 3-DSPHB algorithm, destinations are divided into six subsets. A source node may send a message through six ports simultaneously. In 3-DHB algorithm 3-D mesh are divided into two subsets. However, a potential disadvantage of 3-DSPHB routing is not revealed until the load is relatively high. The source node in 3-DSPHB algorithm will send the messages from all outgoing channels (here is six channels) to reach comparatively large set of destinations, the source node will likely send on all of its outgoing channels. If 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 throttle system throughput and increase message latency. This throttle case is less likely to occur in 3-DHB routing,

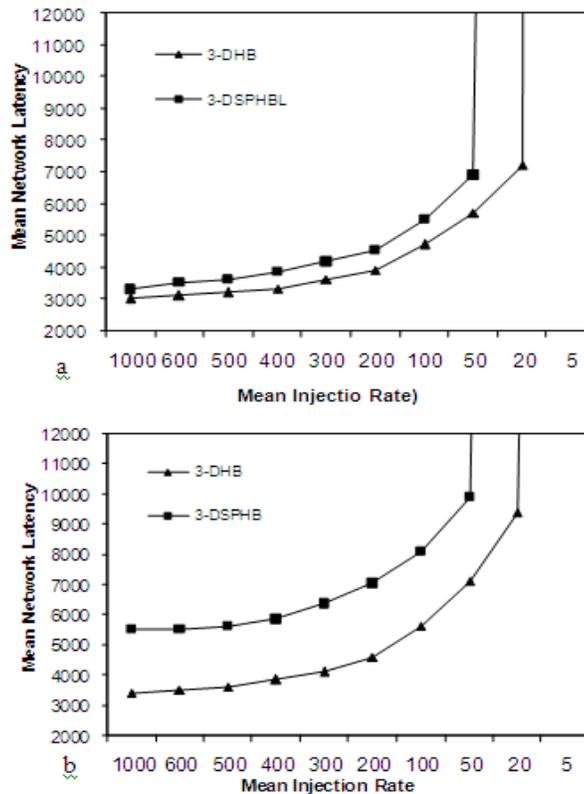


Fig. 9 Comparison Of 3-DHB With 3-DSPHB. (A) With Small Message Latency $\beta = 10$; (B) With Large Message Latency $\beta = 100$

6. 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 two algorithms for broadcasting messages on all-port wormhole-routed 3-D mesh with arbitrary size. These algorithms are shown to be deadlock-free. The performance study shows that the 3-DHB algorithm outperforms the 3-DSPHB algorithm for different length of messages and different injection rate with reduced delays. The broadcast latency of the 3-DSPHB algorithm linearly increases as message size and injection rate. The number of messages that waiting to be delivered in the node's memory is linearly increase with the number of ports provided in the wormhole-routed, so the algorithm that contains a large number of ports is average latency increases.

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] W. Dally and C. Seitz, (1986) "The Torus Routing Chip", J. Distributed Computing, Vol. 1, No. 3, pp 187-196.
- [10] L.M. Ni and P.K. McKinley, (1993) "A Survey of Wormhole Routing Techniques in Directed Network", Computer, Vol. 26, No. 2, pp 62-76.
- [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 & Distrubted 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] X. Lin, P.K. McKinley, L.M. Ni, (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.
- [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] J. Duato, C. Yalamanchili, L. Ni, (2003) “Interconnection Networks: An Engineering Approach”, Elsevier Science.
- [20] M.P. Malumbres, J. Duato, (2000) “An efficient implementation of tree-based multicast routing for distributed shared-memory multiprocessors”, *Journal of Systems Architecture*, Vol. 46, pp 1019–1032.
- [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] 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.
- [24] Tim S. Axelrod, (1986) “ Effects of synchronization barriers on multiprocessor performance”, *Parallel Computing*, Vol. 3, No. 2, pp 129-140.
- [25] 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.
- [26] 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.
- [27] 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*.
- [28] 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
- [29] Z. Shen, (2007) “ A generalized broadcasting schema for the mesh structures”, *Applied Mathematics and Computation*, Vol. 186, No. 2, pp 1293–1310.
- [30] 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.
- [31] 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.
- [32] 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.
- [33] Ni L.M, McKinley P.K, (1993) “A survey of Wormhole Routing Techniques in Direct Networks” [J]. *IEEE Computer*, Vol. 26, No. 2, pp 62-76.
- [34] Lin X, McKinley P.K, Ni L.M, (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.
- [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] H. D. Schwetman, (1985) “CSIM: A C-based, process-oriented simulation language, Tech. Rep.” *Microelectronics and Computer Technology Corp*, PP 80-85.
- [37] 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. Networks (MASCOTS 93)*, pp 57-62.

AUTHORS

Amnah Elobaid received her B.Sc. degree in Computer Sciences from Faculty of Science, King Abdul Aziz University, Saudi 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.