

X-Hamiltonian Surface Broadcast Algorithm

Amnah El-Obaid

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

Email: dramna@zuj.edu.jo

Received 16 April 2016; accepted 25 June 2016; published 28 June 2016

Copyright © 2016 by author and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Broadcast is one of the most important approach in distributed memory parallel computers that is used to find a routing approach from 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 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 algorithm for broadcasting on an all-port wormhole-routed 3D mesh with arbitrary size. 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. In this paper, we introduce an efficient algorithm, X-Hamiltonian Surface Broadcast (X-HSB) which uses broadcast communication facility with deadlock-free wormhole routing in general three dimensional networks. In this paper, the behaviors of this algorithm are compared to the previous results using simulation; our paradigm reduces broadcast latency and is simpler. The results presented in this paper indicate the advantage of our proposed algorithm.

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 of 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 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].

Wormhole routing has been widely accepted as the predominant switching mechanism in current multicomputer systems. 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 characterizes with low communication latency [10]. In wormhole routing, each message is divided into packets; the packets are serialized into a sequence of flits (flow control units) for transmission. A flit is the smallest unit of information that a channel can accept or refuse [11]. The header flit of a message governs the route, and the remaining flits follow it in a pipeline fashion through the network. Only the header flit knows where the packet is going, and the remaining data flits must follow the header flit. If all the channels which the header is routed are busy, the header is blocked until one of those channels is freed; the flow control within the network blocks the trailing flits. Wormhole routing avoids using storage bandwidth in the nodes through which messages are routed. It also makes the message latency quite insensitive to the destination in the network. 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]-[25].

The main facility of the wormhole routing is the communication latency, where it is 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].

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 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. Many algorithms have been proposed for broadcast communication in wormhole-routed networks over the past few years [28]-[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) \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:

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

$$E(M) = \left\{ \left[(x_i, y_i, z_i), (x_j, y_j, z_j) \right] \mid (x_i, y_i, z_i), (x_j, y_j, z_j) \in V(G), \right. \\ \left. \text{and } |x_i - x_j| + |y_i - y_j| + |z_i - z_j| = 1 \right\}$$

Three kinds of ports can be distinguished in mesh topology. One-Port, Multi-Ports and All-Ports [33]. 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 [34] and [35].

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 introduce the new broadcast algorithm based on the Hamiltonian model. In Section 4 we examine and compare the performance of our proposed algorithm with other previous algorithms. Finally, Section 5 concludes this study.

2. Related Works

Interconnection networks are used in massively parallel computers to allow cooperation between several nodes (processing elements). An important advantage of Hamiltonian paths is to design a deadlock-free algorithm of wormhole-routed network. The basic function of an interconnection network is the routing algorithm. Routing Algorithm determines the path from a source node to a destination node in a particular network. In this section we produce two routing algorithm depends in Hamiltonian paths that are presented in [36].

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. In 3-DHB 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 [36].

In 3-DHB, 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 [36].

To study the performance of 3-DHB algorithm, consider the example shown in **Figure 1** 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.

The 3-DSPHB algorithm for All-Port 3-D mesh based on the Hamiltonian paths to implement broadcast in six message-passing steps. A major consequence of All-Port architecture is that the local processor may transmit (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 same partitioned schema can be done for set D_L [36].

To study the performance of 3-DSPHB algorithm, consider the example shown in **Figure 2** 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.

3. The Proposed Works

3.1. Hamiltonian Models to the 3-D Mesh Networks as Set of Surfaces

Hamiltonian-path routing is considered as one of the most efficient important approach for designing a

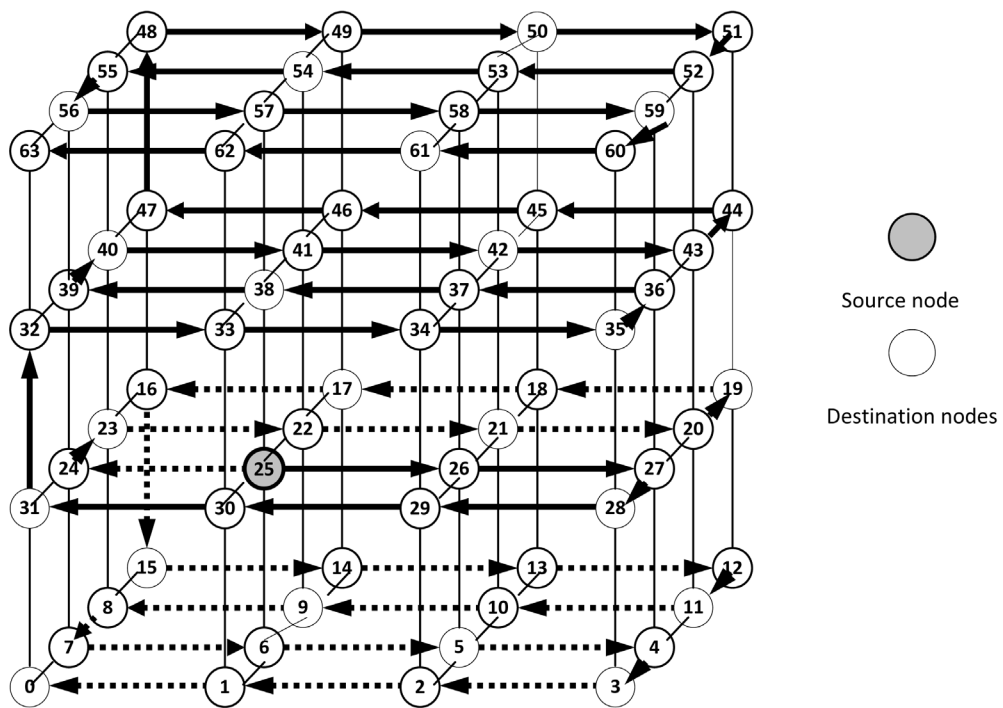


Figure 1. The routing patterns of 3-DHB algorithm, bold lines of N_U subnetwork, and dashed lines of N_L subnetwork.

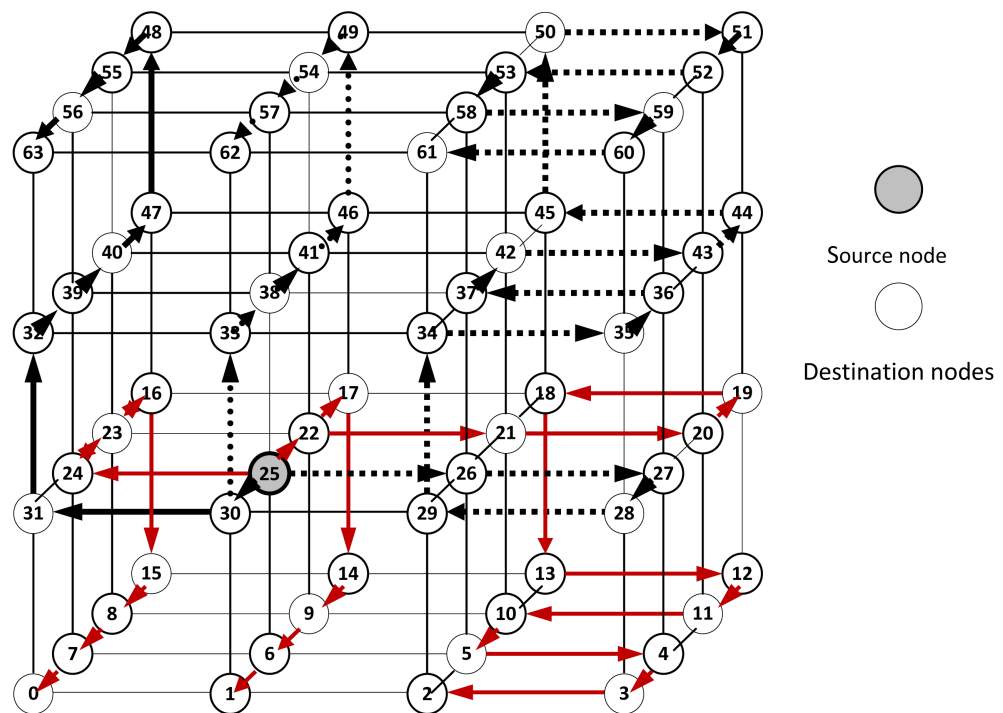


Figure 2. The routing patterns of 3-DSPHB algorithm.

deadlock-free algorithm of wormhole-routed network [37] and [38]. In this paper the routing algorithm is the Hamiltonian-path, 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 can be divided into set of column surfaces. Each surface will be labeled with Hamilton model. 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 assignment a label function ℓ to each surface (according to x dimension) of 3-D mesh which can be expressed in terms of the x -, y - and z -coordinates as follows:

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

Each node in each surface can be assign a unique number using the above labeling function starts from 0, followed by 1, and so on until the last node labeled $n * r - 1$.

Using the Hamiltonian model labeling, each surface (set of column surface) can be partitioned into two sub-networks. The first one is called high-channel network (N_U) 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 (N_L) and contains all channels where their routing directions are from the nodes labeled from high to low numbers. According to this partition schema, the 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 $3 \times 3 \times 3$ 3-D mesh can be shown in **Figure 3(a)**, where every node can be represented by its integer coordinate (x, y, z) and there are three surfaces of subnet works according to x coordinates. A high-channel subnetwork (N_U) can be shown in **Figure 3(b)** for each surface. A low-channel subnetwork (N_L) can be shown in **Figure 3(c)** for each surface.

3.2. The Proposed Algorithm: X-Hamiltonian Surface Broadcast (X-HSB)

In this paper our algorithm X-HSB algorithm is based on splitting the 3-D mesh network into set of surfaces. Each surface will be labeled with Hamilton model that presented in Section 3.1. In fact, if we have $m \times n \times r$ 3-D mesh, then we have m surfaces of mesh. Formally, the sub-networks are described by the following expression:

$$\begin{aligned} N_{x_0} &= \{(x, y, z) | (x = 0), (0 \leq y < n), (0 \leq z < r)\}, \\ N_{x_1} &= \{(x, y, z) | (x = 1), (0 \leq y < n), (0 \leq z < r)\}, \\ &\vdots \\ N_{x_{m-1}} &= \{(x, y, z) | (x = m - 1), (0 \leq y < n), (0 \leq z < r)\} \end{aligned}$$

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 2 subsets D_S and D_R , where D_S contain the nodes which their xcoordinates are equal x_0 , D_R contain the rest nodes which their x coordinates are not equal x_0 . Formally, the subsets are described by the following expression:

$$\begin{aligned} D_U &= \{(x, y, z) | (x, y, z) \in D_S, \ell(x, y, z) > \ell(x_0, y_0, z_0)\} \\ D_L &= \{(x, y, z) | (x, y, z) \in D_S, \ell(x, y, z) < \ell(x_0, y_0, z_0)\} \end{aligned}$$

Step 2: The source node divides D_S into two subsets D_U and D_L as follows:

$$\begin{aligned} D_U &= \{(x, y, z) | (x, y, z) \in D_S, \ell(x, y, z) > \ell(x_0, y_0, z_0)\} \\ D_L &= \{(x, y, z) | (x, y, z) \in D_S, \ell(x, y, z) < \ell(x_0, y_0, z_0)\} \end{aligned}$$

Step 3: The source sort the nodes in D_U in ascending order and the destination nodes in D_L in descending order as the label key.

Step 4: 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 .

Step 5: The source divides D_R into 2 subsets D_{RXG} and D_{RXS} , where D_{RXG} contain the nodes which their x

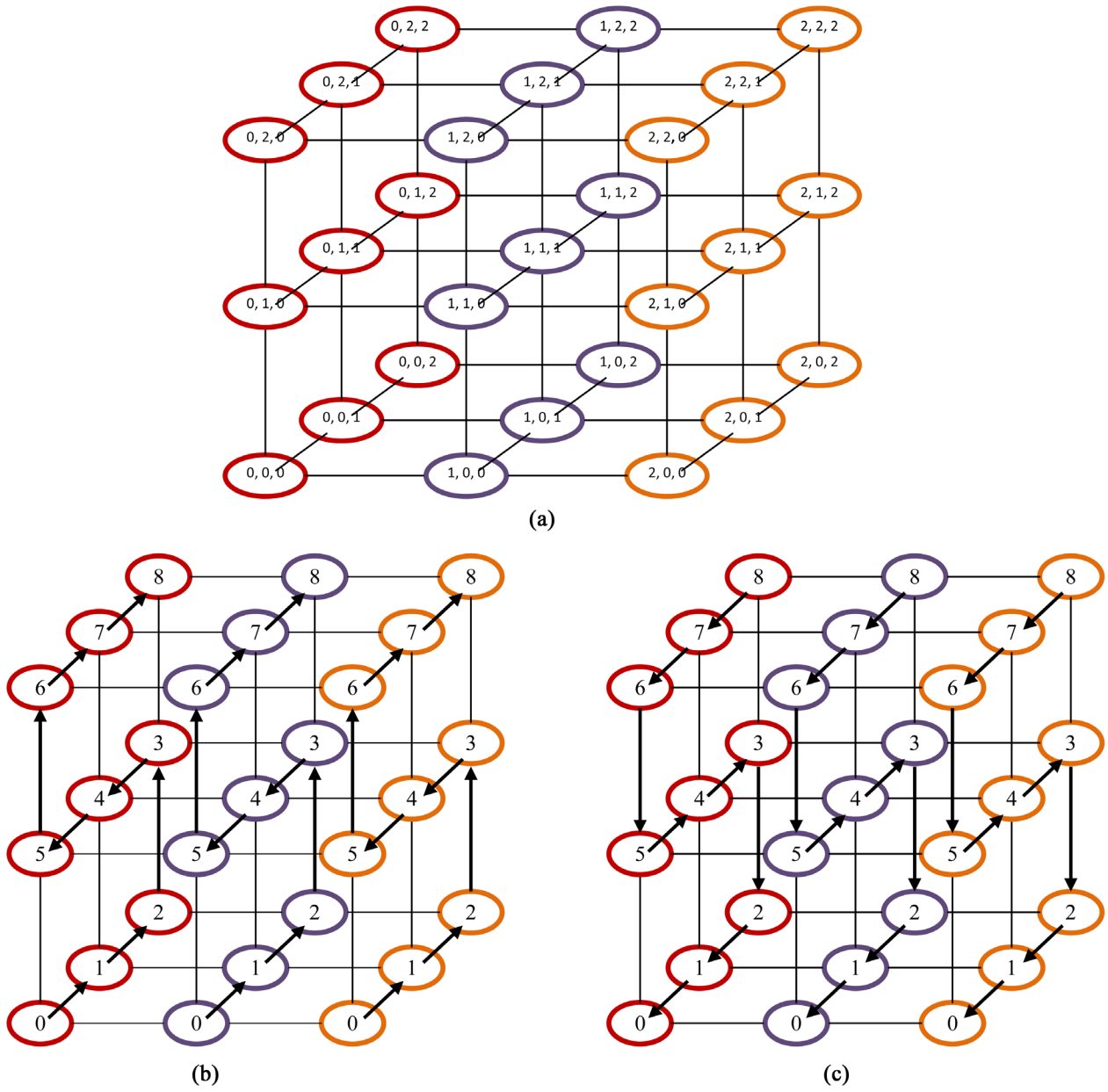


Figure 3. The labeling of a $3 \times 3 \times 3$ mesh. (a) Physical network. (b) High-channel network surfaces (N_U). (c) Low-channel network surfaces (N_L).

coordinates are greater than x_0 and D_{RXS} contain the nodes which their x coordinates are smaller than x_0 .

Step 6: The source divides D_{RXG} into set of subsets. Formally, the subsets of D_{RXG} are described by the following expression:

$$\begin{aligned}
 D_{R_{x_0+1}} &= \{(x, y, z) \mid (x, y, z) \in D_{RXG}, (x = x_0 + 1), (0 \leq y < n), (0 \leq z < r)\}, \\
 D_{R_{x_0+2}} &= \{(x, y, z) \mid (x, y, z) \in D_{RXG}, (x = x_0 + 2), (0 \leq y < n), (0 \leq z < r)\}, \\
 &\vdots \\
 D_{R_{m-1}} &= \{(x, y, z) \mid (x, y, z) \in D_{RXG}, (x = m - 1), (0 \leq y < n), (0 \leq z < r)\}
 \end{aligned}$$

Step 7: Each above subset of D_{RXG} represents a surface. The source (x_0, y_0, z_0) sends the message to a node,

which is represented by its integer coordinate (x_{0+1}, y_0, z_0) which will be act as a source on its surface, to a node represented by its integer coordinate (x_{0+2}, y_0, z_0) which will be act as a source on its surface and so on to a node represented by its integer coordinate (x_{m-1}, y_0, z_0) which will be act as a source on its surface. All the alternative sources will send the message to all nodes in their surfaces by repeating the above steps from step 2 to step 4.

Step 8: The source divides D_{RXS} into set of subsets. Formally, the subsets of D_{RXS} are described by the following expression:

$$\begin{aligned} D_{R_{x0-1}} &= \{(x, y, z) \mid (x, y, z) \in D_{RXS}, (x = x_0 - 1), (0 \leq y < n), (0 \leq z < r)\}, \\ D_{R_{x0-2}} &= \{(x, y, z) \mid (x, y, z) \in D_{RXS}, (x = x_0 - 2), (0 \leq y < n), (0 \leq z < r)\}, \\ &\vdots \\ D_{R_0} &= \{(x, y, z) \mid (x, y, z) \in D_{RXS}, (x = 0), (0 \leq y < n), (0 \leq z < r)\} \end{aligned}$$

Step 9: Each above subset of D_{RXS} represents a surface. The source (x_0, y_0, z_0) sends the message to a node, which is represented by its integer coordinate (x_{0-1}, y_0, z_0) which will be act as a source on its surface, to a node represented by its integer coordinate (x_{0-2}, y_0, z_0) which will be act as a source on its surface and so on to a node represented by its integer coordinate $(0, y_0, z_0)$ which will be act as a source on its surface. All the alternative sources will send the message to all nodes in their surfaces by repeating the above steps from step 2 to step 4.

Theorem 1: X-HSB algorithm is deadlock-free.

Proof: At the source node, X-HSB algorithm divides the network into m disjoint subnetworks. This is obvious since $N_{x_0} \cap N_{x_1} \cap N_{x_2} \cap \dots \cap N_{x_{m-1}} = \phi$. Then X-HSB algorithm is deadlock-free at each subnetwork. To complete our proof, we should prove that there are no dependencies in each subnetwork. Each subnetwork will be divided into two disjoint subnetworks 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 X-HSB is deadlock-free.

3.3. Comparative Study

To study the performance of X-HSB algorithm, consider the example shown in **Figure 4** for $4 \times 4 \times 4$ mesh topology labeling using a Hamiltonian path for each surface. The source node with Hamilton labeled 5 at surface 2 which has an integer coordinate $(1, 1, 2)$ will send a broadcast to all nodes in its 3-D mesh multicomputer.

The routing pattern of X-HSB algorithm is shown with bold lines as shown in **Figure 4** for each surface, where first surface have x coordinate = 0, second surface have x coordinate = 1, third surface have x coordinate = 2 and fourth surface have x coordinate = 3.

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 [39]. 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 [40]. MultiSim was used to simulate broadcast operations in 3D mesh of different sizes.

To study the performance of our algorithm 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 $6 \times 6 \times 6$ 3-D mesh. The software that overhead for buffers allocating, messages coping, router initializing, etc. is represented the startup latency β . 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 X-HSB, 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 β is set to 100 microseconds are shown in **Figure 5**. The advantage of the X-HSB algorithm is peter performance than the advantage of the 3-DHB and 3-DSPHB algorithms. In X-HSB the source node divided the net into m subnets (no. of columns) as shown in Section 3.1 and

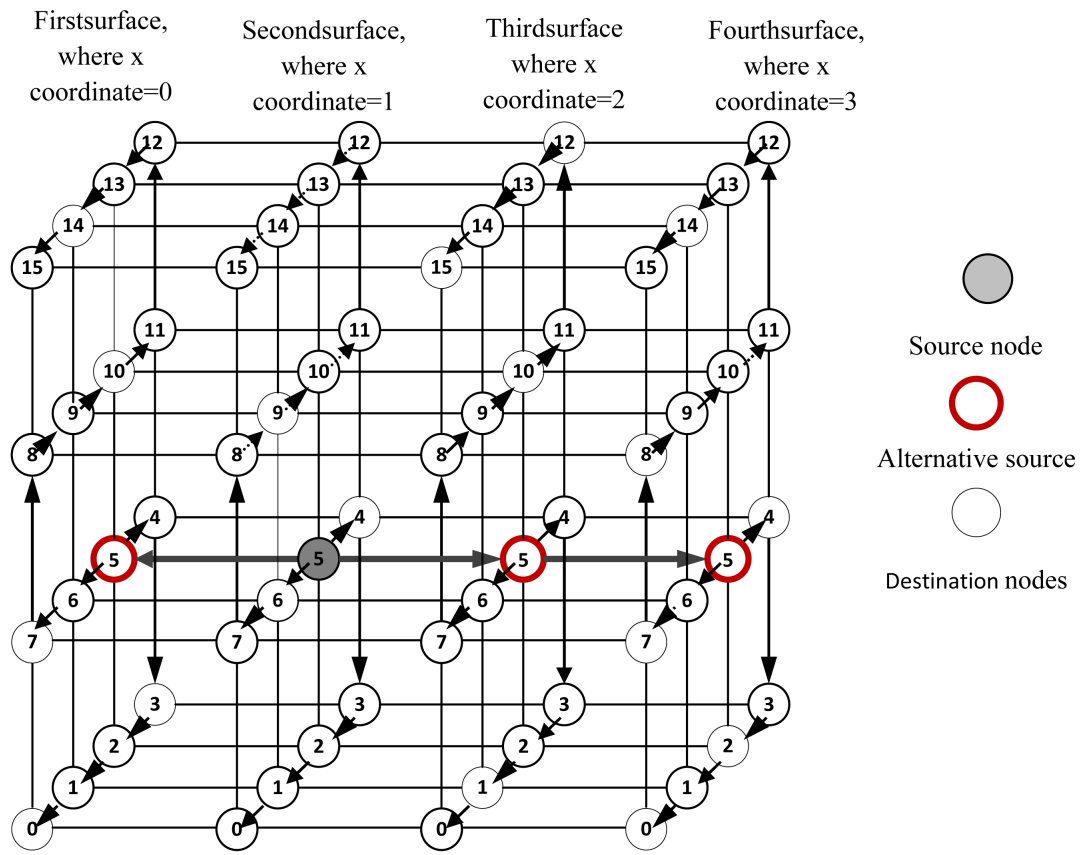


Figure 4. The routing patterns of X-HSB algorithm.

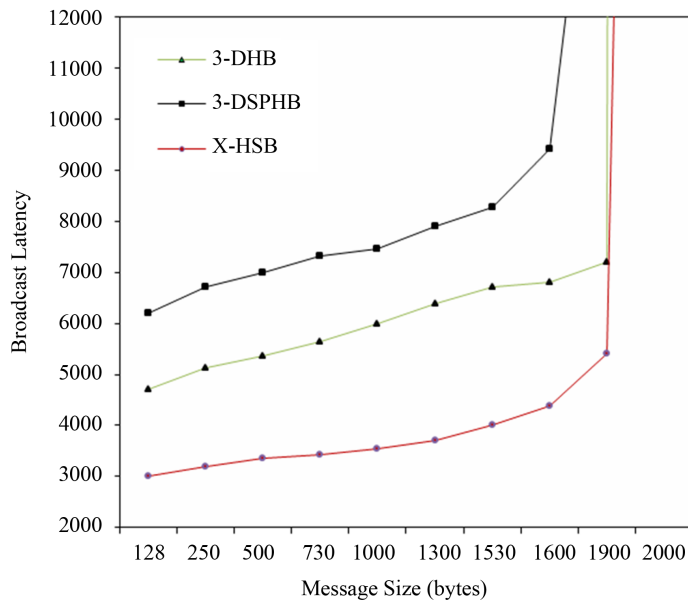


Figure 5. Performance comparison of X-HSB algorithm with 3-DHB and 3-DSPHB algorithms under different message size.

each subnets divided their destinations into two subsets (use Hamilton-path for each surface), 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 algo-

rithms; this advantage can be shown in terms of generated traffic. Since communication latency in wormhole-routed broadcast systems is dependent on the length of the message, average latencies of X-HSB algorithm is less sensitive as the length of the message as shown in Figure 5. The performance depends strongly on the start-up time that 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 average latencies of 3-DSPHB algorithm linearly increase as the length of the message. However, the disadvantage of 3-DSPHB algorithm increases with the message lengths.

Figure 6 plot the network latency obtained by the three algorithms versus various network loads. The startup latency β is set to 100 microseconds. We have fixed the message length as 300 flits. All three algorithms exhibit good performance at low loads. The X-HSB algorithms, however, are less sensitive to increased load than the 3-DHB and 3-DSPHB algorithms. This result occurs because in the X-HSB algorithm the source node divide the destination nodes into 2 subset for each surface (no. of coulombs), 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. The X-HSB routing exhibits lower latency than 3-DHB and 3-DSPHB routing because, as shown earlier, paths tend to be shortest, generating less traffic. Hence, the network will not saturate as quickly. The disadvantage of 3-DHB and 3-DSPHB routing algorithms increases when loads on net is very high.

5. Conclusions

Broadcast, as one of the most fundamental collective communication operations, is highly demanded in massive parallel computer algorithms like parallel graph algorithms, fast Fourier transform, and barrier synchronization. Broadcast is frequently invoked as one of the most essential communication operations.

In this paper, we propose a widely applicable solution for the broadcast problem in a wormhole routed mesh. We propose a new broadcasting algorithm (X-HSB) which guarantees the delivery (to all processors connected to the source) in the 3-D meshes. Our algorithms are proposed with variants that can be easily expressed and understood. We have also shown that our proposed algorithm is deadlock-free. The performance study shows that the proposed algorithm X-HSB is the best performance than the existing algorithms 3-DHB and 3-DSPHB.

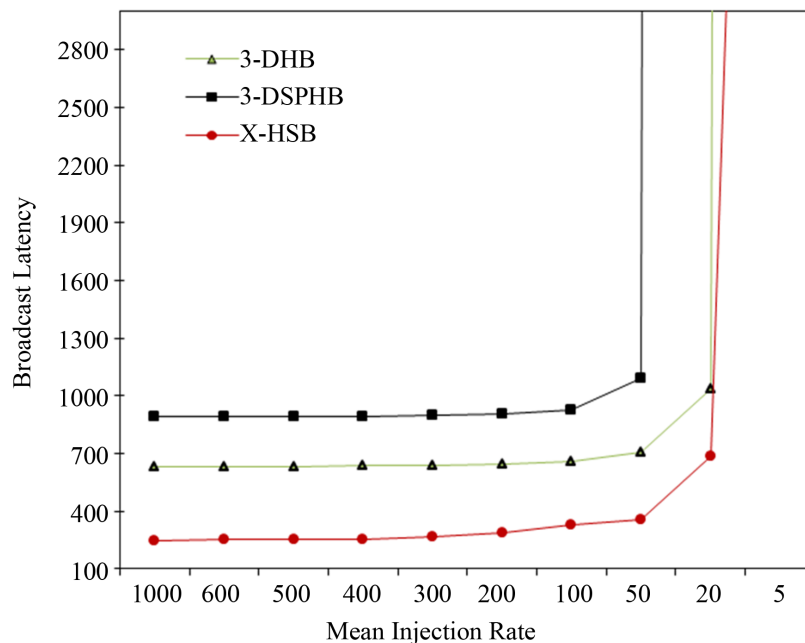


Figure 6. Performance comparison of X-HSB algorithm with 3-DHB and 3-DSPHB algorithms under different loads.

The X-HSB algorithm is less sensitive to message length and injection rate. 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.

References

- [1] Khan, M.Y., Tyagi, S. and Khan, M.A. (2014) Tree-Based 3-D Topology for Network-on-Chip World. *Applied Sciences Journal*, **30**, 844-851.
- [2] Intel Corporation (1990) A Touchstone DELTA System Description. Intel Corporation, Intel Supercomputing Systems Division.
- [3] Nuth, P.R. and Dally, W.J. (1992) The J-Machine Network. *IEEE International Conference on Computer Design: VLSI in Computers and Processors*, **3**, 420-423.
- [4] Foschia, R., Rauber, T. and Runger, G. (1997) Modeling the Communication Behavior of the Intel Paragon. *Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*, IEEE Computer Society Press, 117-124. <http://dx.doi.org/10.1109/mascot.1997.567594>
- [5] Almasi, G.S. and Gottlieb, A. (1994) Highly Parallel Computing Benjamin/Cummings.
- [6] Athas, W.C. and Seitz, C.L. (1988) Multicomputers Message Passing Concurrent Computers. *IEEE Comp*, **21**, 9-24. <http://dx.doi.org/10.1109/2.73>
- [7] Lessler, R.E. and Schwazmeier, J.L. (1993) CRAY T3D: A New Dimension for Cray Research in COMPCON. *IEEE Computer Society Press*, **8**, 176-182.
- [8] Cray Research Inc. (1995) CRAY T3E Scalable Parallel Processing System. Cray Research Inc. <http://www.cray.com/products/systems/crayt3e/>
- [9] Dally, W. and Seitz, C. (1986) The Torus Routing Chip. *Journal of Parallel and Distributed Computing*, **1**, 187-196. <http://dx.doi.org/10.1007/BF01660031>
- [10] Karkar, A., Dahir, N., Al-Dujaily, R., Tong, K., Mak, T. and Yakovlev, A. (2014) Hybrid Wire-Surface Wave Architecture for One-to-Many Communication in Networks-on-Chip. *Journal of Parallel and Distributed Computing*, **61**, 1307-1336.
- [11] Dally, W.J. and Towles, B. (2004) Principles and Practices of Interconnection Networks. Morgan Kaufmann Publishers, 13.2.1.
- [12] Samman, F.A. (2011) New Theory for Deadlock-Free Multicast Routing in Wormhole-Switched Virtual Channel Less Networks On-Chip. *IEEE Transactions on Parallel & Distributed System*, **22**, 544-557. <http://dx.doi.org/10.1109/TPDS.2010.120>
- [13] Omari, M. (2014) Adaptive Algorithms for Wormhole-Routed Single-Port Mesh Hypercube Network. *JCSI International Journal of Computer Science Issues*, **11**, 1694-0814.
- [14] Moharam, H., Abd El-Baky, M.A. and Yomna, S.M.M. (2000) An Efficient Deadlock Free Multicast Wormhole Algorithm in 2-D Mesh Multicomputers. *Journal of systems Architecture*, **46**, 1073-1091. [http://dx.doi.org/10.1016/S1383-7621\(00\)00010-2](http://dx.doi.org/10.1016/S1383-7621(00)00010-2)
- [15] Wang, N.-C., Chu, C.-P. and Chen, T.-S. (2002) A Dual Hamiltonian-Path-Based Multicasting Strategy for Wormhole Routed Star Graph Interconnection Networks. *Journal of Parallel and Distributed Computing*, **62**, 1747-1762. [http://dx.doi.org/10.1016/S0743-7315\(02\)00007-2](http://dx.doi.org/10.1016/S0743-7315(02)00007-2)
- [16] Lin, X., McKinley, P.K. and Ni, L.M. (1994) Deadlock-Free Multicast Wormhole Routing in 2-D Mesh Multicomputers. *IEEE Transactions on Parallel and Distributed Systems*, **5**, 793-804. <http://dx.doi.org/10.1109/71.298203>
- [17] Fleury, E. and Fraigniaud, P. (1998) Strategies for Path-Based Multicasting in Wormhole-Routed Meshes. *Journal of Parallel & Distributed Computing*, **6**, 26-62. <http://dx.doi.org/10.1006/jpdc.1998.1473>
- [18] McKinley, P., Tsai, Y.J. and Robinson, D. (1995) Collective Communication in Wormhole-Routed Massively Parallel Computers. *IEEE Computer*, **28**, 39-50. <http://dx.doi.org/10.1109/2.476198>
- [19] Matam, R. and Tripathy, S. (2013) WRSR: Wormhole-Resistant Secure Routing for Wireless Mesh Networks. *EURASIP Journal on Wireless Communications and Networking*, **2013**, 180. <http://dx.doi.org/10.1186/1687-1499-2013-180>
- [20] Karlsson, J., Dooley, L.S. and Pulkkis, G. (2013) Identifying Time Measurement Tampering in the Traversal Time and Hop Count Analysis (TTHCA) Wormhole Detection Algorithm. *Sensors (Basel)* **13**, 6651-6668. <http://dx.doi.org/10.3390/s130506651>
- [21] Kumar, D.R., Najjar, W.A. and Srimani, P.K. (2001) A New Adaptive Hardware Tree-Based Multicast Routing in K-Ary N-Cubes. *IEEE Transactions on Computers*, **50**, 647-659. <http://dx.doi.org/10.1109/12.936232>
- [22] Fan, J.X. (2002) Hamilton-Connectivity and Cycle-Embedding of the Mobius Cubes. *Information Processing Letters*,

- 82**, 113-117. [http://dx.doi.org/10.1016/S0020-0190\(01\)00256-3](http://dx.doi.org/10.1016/S0020-0190(01)00256-3)
- [23] El-Obaid, A. (2015) Broadcast Wormhole-Routed 3-D Mesh Networks. *International Journal of Computer Networks & Communications (IJCNC)*, **7**, 153-167. <http://dx.doi.org/10.5121/ijcnc.2015.7411>
- [24] Div. of Math. & Comput. Sci., Texas Univ., San Antonio, TX, USA (1994) An Efficient Path-Based Multicast Algorithm for Minimum Communication Step. *Proceedings of 6th IEEE Symposium on Parallel and Distributed Processing*, **7**, 722-729.
- [25] Ruiz, P. (2015) Survey on Broadcast Algorithms for Mobile Ad Hoc Networks. *ACM Computing Surveys (CSUR)* **48**, Article Number: 8.
- [26] Hamed, K. and El-Sayed, M.A. (2015) BTL—An Efficient Deadlock-Free Multicast Wormhole Algorithm to Optimize Traffic in 2D Torus Multicomputer. *International Journal of Computer Applications*, **111**, 32-37. <http://dx.doi.org/10.5120/19546-1415>
- [27] Axelrod, T.S. (1986) Effects of Synchronization Barriers on Multiprocessor Performance. *Parallel Computing*, **3**, 129-140. [http://dx.doi.org/10.1016/0167-8191\(86\)90030-X](http://dx.doi.org/10.1016/0167-8191(86)90030-X)
- [28] Wang, H. and Wu, L. (2012) Preconcerted Wormhole Routing Algorithm for Mesh Structure Based on the Network on Chip, Information Management, Innovation Management and Industrial Engineering (ICIII). 2012 *International Conference on Information Management, Innovation Management and Industrial Engineering*, Vol. 2, Sanya, 20-21 October 2012, 154-158. <http://dx.doi.org/10.1109/ICIII.2012.6339801>
- [29] Chen, Y.-S. and Lin, Y.-C. (2001) A Broadcast-VOD Protocol in an Integrated Wireless Mobile Network. *Journal of Internet Technology*, **2**, 143-154.
- [30] Moadeli, M. and Vander Bauwhede, W. (2009) A Communication Model of Broadcast in Wormhole-Routed Networks on-Chip. *International Conference on Advanced Information Networking and Applications*, Bradford, 26-29 May 2009, 315-322. <http://dx.doi.org/10.1109/AINA.2009.126>
- [31] Seo, J.-H. and Lee, H.O. (2013) Link-Disjoint Broadcasting Algorithm in Wormhole-Routed 3D Petersen-Torus Networks. *International Journal of Distributed Sensor Networks*, **2013**, Article ID: 501974. <http://dx.doi.org/10.1155/2013/501974>
- [32] Shen, Z. (2007) A Generalized Broadcasting Schema for the Mesh Structures. *Applied Mathematics and Computation*, **186**, 1293-1310. <http://dx.doi.org/10.1016/j.amc.2006.07.153>
- [33] Seo, J.-H. (2013) Three-Dimensional Petersen-Torus Network: A Fixed-Degree Network for Massively Parallel Computers. *Journal of Supercomputing*, **64**, 987-1007. <http://dx.doi.org/10.1007/s11227-011-0716-z>
- [34] Li, Y., Peng, S. and Chu, W. (2012) Hierarchical Dual-Net: A Flexible Interconnection Network and its Routing Algorithm. *International Journal of Networking and Computing*, **2**, 234-250.
- [35] Anand, V., Sairam, N. and Thiyagarajan, M. (2012) A Review of Routing in Ad Hoc Networks. *Research Journal of Applied Sciences, Engineering and Technology*, **4**, 981-986.
- [36] El-Obaid, A. (2015) Three-Dimension Hamiltonian Broadcast Wormhole-Routing. *International Journal of Computer Networks & Communications (IJCNC)*, **7**, 31-46. <http://dx.doi.org/10.5121/ijcnc.2015.7303>
- [37] El-Obaid, A. and Zuo, W.-L. (2008) An Efficient Path-Based Multicast Algorithm for Minimum Communication. *Information Technology Journal*, **7**, 32-39. <http://dx.doi.org/10.3923/itj.2008.32.39>
- [38] El-Obaid, A. and Zuo, W.-L. (2007) Hamiltonian Paths for Designing Deadlock-Free Multicasting Wormhole-Routing Algorithms in 3-D Meshes. *Journal of Applied Sciences*, **7**, 3410-3419. <http://dx.doi.org/10.3923/jas.2007.3410.3419>
- [39] Schwetman, H.D. (1985) CSIM: A C-Based, Process-Oriented Simulation Language. Technical Report, Microelectronics and Computer Technology Corp, 80-85.
- [40] McKinley, P.K. and Trefftz, C. (1993) MultiSim: A Tool for the Study of Large-Scale Multiprocessors. *Proceedings of International Workshop on Modeling, Analysis and Simulation of Computer and Telecommunications Systems (MASCOTS 93)*, San Diego, 17-20 January 1993, 57-62.