Y-HAMILTONIAN LAYERS BROADCAST ALGORITHM

Amnah El-Obaid[1] and Nagham_Al-Madi[2]
[1]Department of Basic Science, Faculty of Science and Information Technology, Al-Zaytoonah University of Jordan, Amman, Jordan
[2]Department of Computer Science, Faculty of Science and Information Technology, Al-Zaytoonah University of Jordan, Amman, Jordan

ABSTRACT:

Anew approach to broadcast in wormhole routed three-dimensional networks is proposed. One of the most important process in communication and parallel computer is broadcast approach.. 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. 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. Wormhole routing is divided the packets into set of FLITS (flow control digits). The first Flit of the packet (Header Flit) is containing the destination address and all subsets flits will follow the routing way of the header Flit. In this paper, we consider an efficient algorithm for broadcasting on an all-port wormhole-routed 3D mesh with arbitrary size. We introduce an efficient algorithm, Y-Hamiltonian Layers Broadcast(Y-HLB). 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

Multicomputer architecture is one of the most and important approach that is used in the computation and high speed computing algorithms. The communications in multicomputer system is very important for several reasons; the first one is to exchange or event synchronization. The second one efficient communication has been recognized to be critical for high performance computing. The topology of Multicomputer system can be consists of several nodes that are connected together in a special way each of node consist of microprocessor and local memory. The n-dimensional meshes or k-ary n-cubes is the most topology that is used in multicomputer networks because their regular topologies and simple routing. According to their topologies, the meshes, tori, hypercubes and trees is the most commonly used in multicomputer network. The implementation of the mesh topology is very easy and very simple, so they used easy to understand and used. 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].

One of the most widely used in multicomputer systems is Wormhole routing.. One of the efficient message routing algorithms of multicomputers is wormhole-switching. The routing algorithms used wormhole routing widely because of some reasons, firstly the communication latency that required to buffering a message is very low [9]. Secondly, it characterized with low communication latency [10]. In wormhole routing, each message is divided into packets; the

DOI: 10.5121/ijnsa.2016.8303

packets are serialized into a sequence of flits (flow control units) for transmission. The channel can accept or refuse this flits which is the smallest unit of packets [11]. The header flit of a message governs the route, and the remaining flits follow it in a pipeline fashion through the network. All destination addresses information of the message will be stored in the header flit, so the header flits is going and the remaining flits follow it. When the header flit rout in a channel and this channel is busy then the header is blocked until one of those channels is freed; the routing stream within network will be blocked the trailing flits. Wormhole routing avoids using storage bandwidth in the nodes through which messages are routed. The message latency in Wormhole routing is quite insensitive to the destination nodes in the network. When set of message is blocked because the channel that is required is used by another message then deadlock is occurred. The communication in the network will be blocked when deadlock is occurred, so the communication can be done when exceptional action is taken to break this deadlock [12], [13]. According to these features, The research propose many wormhole routing algorithms in the network communication that can be founded in [14], [15], [16], [17], [18], [19], [20], [21] and [22].

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 time is required to start a message is called the start-up latency, 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 delays that occurred for the message during routing algorithms is called the blocking latency [23].

Broadcast is one of the most basic communication routing algorithm in multicomputer where the source send the message 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 [24], which are supported in data parallel languages. In the past few years many studies produced different algorithms for broadcast communication in wormhole-routed networks [25], [26] [27] and [28]. A broadcasting algorithm is produced 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].

The topology of mesh that used in this paper is 3-D mesh with Bi-directional channels. The 3-D mesh can be modeled as a graph M (V, E) where each node represent a processor in V (M) and each edge represent a communication channel 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: A 3-D mesh graph of m x n x r is a directed graph M (V, E), which is supply the following conditions :-

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

The mesh topology supports three kinds of ports as follow:-. One-Port, Multi-Ports and All-Ports [30]. The All-Port router model where routers can send/receive multiple messages simultaneously to a neighboring node is used in this paper, and that node can simultaneously send/receive messages along all ejection and injection channels [31] and [32].

Hamilton path is the most and important approach that is used for broadcast routing. The new broadcast routing algorithm used Hamiltonian model for 3-D mesh multicomputer is produced in this paper. The rest of the paper is organized as follows. Section 2 presents the related works. The Hamiltonian model to the 3-D mesh networks is presented in Section 3 and also we introduce our new algorithm in this section. 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. The path from a source node to a destination node in a network is called routing Algorithm. In this section we produce two routing algorithm depends in Hamiltonian paths that are presented in [33].

2.1 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 [33] 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 [33].

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 [33].

To study the performance of 3-DHB algorithm, consider the example shown in Fig. 1 for a 4 x 4 x 4 mesh topology labeling using a Hamiltonian path. A broadcast is sent to all nodes in 3-D mesh by the source node that has Hamilton labeled 25 and an integer coordinate (1, 1, 1).

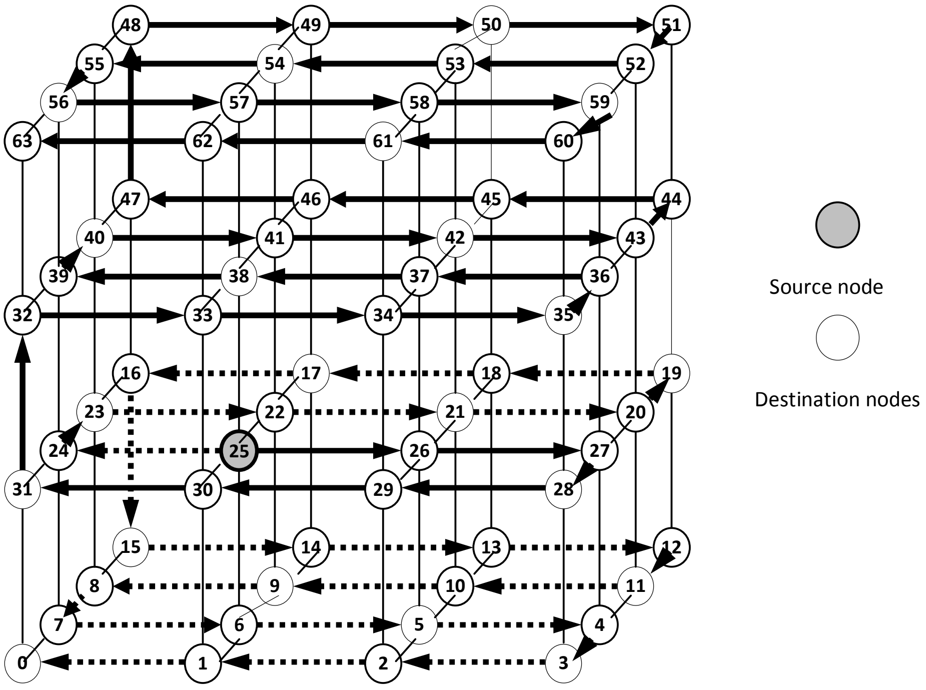

Fig. 1 The routing patterns of 3-DHB algorithm, bold lines of N_U subnetwork, and dashed lines of N_L subnetwork

2.2 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 [33].

To study the performance of 3-DSPHB algorithm, consider the example shown in Fig. 2 for a 4 x 4 x 4 mesh topology labeling using a Hamiltonian path. A broadcast is sent to all nodes in 3-D mesh by the source node that has Hamilton labeled 25 and an integer coordinate (1, 1, 1).

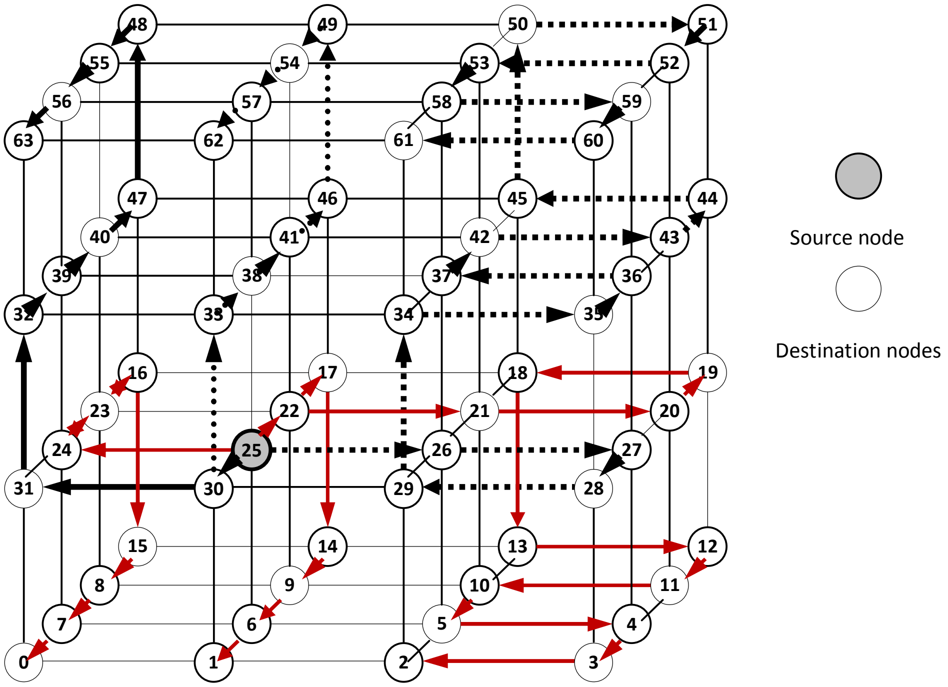

Fig. 2 The routing patterns of 3-DSPHB algorithm

In the GTDTPM algorithm the source divide the network into two subnetworks, N_U and N_L. N_U contains all nodes which their labels are greater than label of the source and N_L have all nodes which their labels are smaller label of the source. The path followed by a message is defined by a deterministic routing function which uses the labeling of the Hamiltonian path. We denote this routing function by R_d. It is defined as a function of the node currently holding a message, and the destination node of this message. The neighboring node which the message should send to it is returns by $R_d : V\,X\,V \rightarrow V$. More specialize, if u represent the current node and v represent the destination node, then R_d (u, v) = w, where w represent the neighboring node of u, and, if L (u) < L (v), then we have the following equation [34].

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

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

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

2.4 General Three-Dimension Binary Tow-Phase Multicast (GTDBTPM)

The main idea of GTDBTPM algorithm is based on splitting the 3-D mesh network into set of layers. We can show the 3-D mesh network as set of layers; where each layer can be represented as layer of 2-D mesh network. Fig. 3 represents 3 x 3 x 3 mesh (each node in the mesh is represented by its integer coordinate (x, y, z)), we can show that there are three layers of 3 x 3 2-D mesh, the z coordinate for the first mesh is 0, for the second mesh is 1, and for the third mesh is 2. Rely for 3-D mesh m x n x r, we have r layers of m x n 2-D mesh. The z coordinate for the first layer is 0, for the second layer is 1, and for the last layer is r-1. In the source the GTDBTPM algorithm will be divided the network according to z coordinate into two subnetworks N_{+z}, and N_{-}

$_z$. The all upper diagonal channels of the source with addresses [(x, y, z), (x, y, z + 1)] is located at Subnetwork N_{+z}, and all lower diagonal channels of source with addresses [(x, y, z), (x, y, z - 1)] is located at subnetwork N_{-z} contains [35]

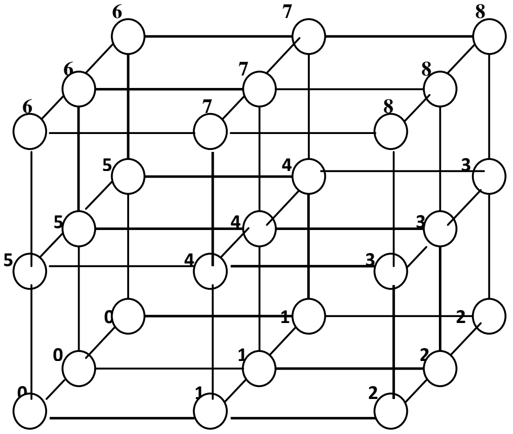

Fig. 3 The routing pattern of GTDBTPM algorithm

3. THE PROPOSED WORKS

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

One of the most important ways for designing deadlock free algorithm using wormhole routing is Hamiltonian-path. So we used Hamiltonian-path routing algorithm in this paper. Our target mesh is 3-D mesh topology using a Hamiltonian path where the messages are routed using the facilities of the Hamiltonian path.

A 3-D mesh network can be divided into set of row layers. We can give each layer of 3-D mesh a label with Hamilton model. In the following, we give the node labeling function $\ell(u)$ for a 3-D mesh. For an m x n x r 3-D mesh, a label function ℓ can be expressed to each Layer (according to y dimension) of 3-D mesh which can be expressed in terms of the x-, y- and z-coordinates as follows:

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

A unique number can be expressed to each node in each layer using the above labeling function starts from 0, followed by 1, and so on until the last node labeled $m*r$-1.

Using the Hamiltonian model labeling, each layer (set of row layers) can be partitioned into two subnetworks. The first one is called high-channel network which consist of 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 which consists of 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 x 3 x 3 3-D mesh can be shown in Fig. 4(a), the integer coordinate (x, y, z) can be used to represent a physical position for every node in 3-D mesh and show us there are three layers of subnet works according to y coordinates. A high-channel subnetwork can be shown in Fig. 4(b) for each layer. A low-channel subnetwork can be shown in Fig. 4(c) for each layer.

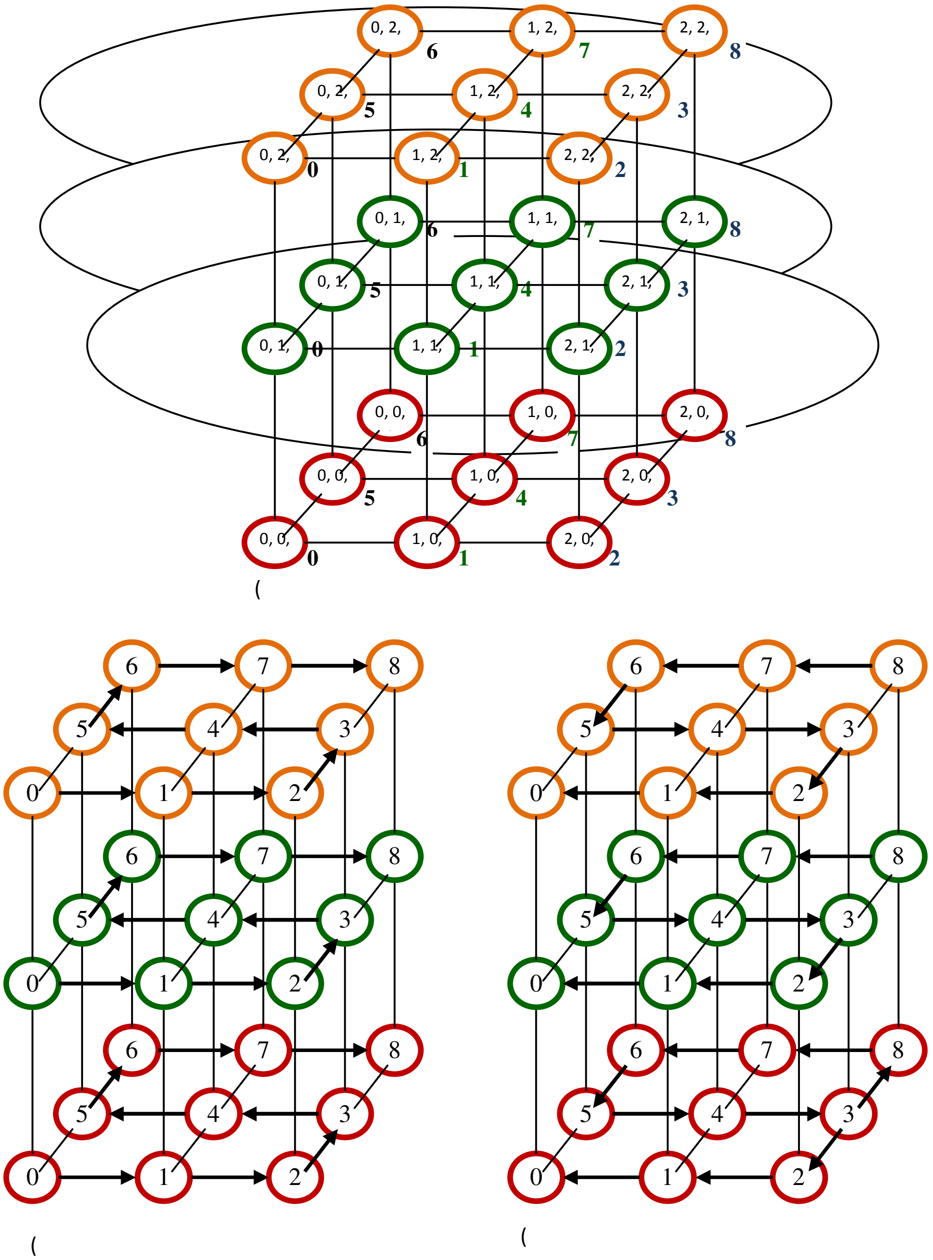

Fig. 4 The labeling of a 3 x 3 x 3 mesh. (a) Physical network. (b) High-channel network layers. (c) Low-channel network layers.

3.2 The proposed algorithm:-Y-Hamiltonian Layers Broadcast(Y-HLB)

The basic idea of our proposed algorithm (Y-HLB) In this paper is based on splitting the 3-D mesh network into set of layers. Each layer will be labeled with Hamilton model that presented in section 3.1. Rely; if we have m x n x r 3-D mesh, then we have n layers of mesh. We can formally the sub-networks by the following expression:

$$N_{y0} = \{(x, y, z)| \ (0 \le x < m), (y = 0), (0 \le z < r)\},$$
$$N_{y1} = \{(x, y, z)| \ (0 \le x < m), (y = 1), (0 \le z < r)\},$$
$$\vdots$$
$$N_{y_{n-1}} = \{(x, y, z)| \ (0 \le x < m), (y = n - 1), (0 \le z < r)\}$$

Suppose the source node named u_0 have the following coordinate (x_0, y_0, z_0), and the rest nodes in the 3-D mesh is named D. 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 y coordinates are equal y_0, D_R contain the rest nodes which their y coordinates are not equal y_0 Formally, the subsets are described by the following expression:

$$D_S = \{(x, y, z)| (x, y, z) \in D, (0 \le x < m), (y = y_0), (0 \le z < r)\},$$
$$D_R = \{(x, y, z)| (x, y, z) \in D, (0 \le x < m), (y \ne y_0), (0 < z < r)\}$$

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

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

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: The source construct two messages, the first one is containing nodes in D_U as part of the header and the second one is containing nodes in D_L as part of the header. The message will be sent to subset D_U through subnetwork N_U by the source and will be sent to subset D_L through subnetwork N_L.

Step 5: The source divides D_R into 2 subsets D_{RYU} and D_{RYL}, where D_{RYU} contain the nodes which their y coordinates are greater than y_0 and D_{RYL} contain the nodes which their y coordinates are smaller than y_0.

Formally, the subsets of D_{RYU} are described by the following expression:

$$D_{R0} = \{(x, y, z)| (x, y, z) \in D_{RYU}, (0 \le x < m), (y = y_0 + 1), (0 \le z < r)\},$$
$$D_{R1} = \{(x, y, z)| (x, y, z) \in D_{RYU}, (0 \le x < m), (y = y_0 + 2), (0 \le z < r \},$$

.

.

.

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

Formally, the subsets of D_{RYL} are described by the following expression:

$$D_{Rn-1} = \{(x, y, z) | (x, y, z) \in D_{RYL}, (0 \le x < m), (y = y_0 - 1), (0 \le z < r) \},$$
$$D_{Rn-2} = \{(x, y, z) | (x, y, z) \in D_{RYL}, (0 \le x < m), (y = y_0 - 2), (0 \le z < r \},$$

.

.

.

$$D_{R0} = \{(x, y, z) | (x, y, z) \in D_{RYL}, (x \le 0 < m), (y = 0), (0 \le z < r) \}$$

Step 6: Each above subset of D_{RYU} represents a layer. The source sends the message to a node, which is represented by its integer coordinate (x_0, y_{0+1}, z_0) which will be act as a source on its layer, to a node represented by its integer coordinate (x_0, y_{0+2}, z_0) which will be act as a source on its layer and so on to a node represented by its integer coordinate (x_0, y_{n-1}, z_0) which will be act as a source on its layer. All the alternative sources will send the message to all nodes in their layers by repeating the above steps from step2 to step 4.

Step 7: Repeat step 6 for subset D_{RYL}.

Theorem 1: Y-HLB algorithm is deadlock-free.

Proof: At the source node, Y-HLB algorithm divides the network into n disjoint subnetworks. This is obvious since $N_{y0} \cap N_{y1} \cap N_{y2} \ldots \cap N_{yn-1} = \phi$. Then Y-HLB algorithm is deadlock-free at each subnetwork. If we proof that there are no dependencies in each subnetwork we will complete our proof. 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 Y-HLB is deadlock-free.

3.3. Comparative study

To study the performance of Y-HLB algorithm, consider the example shown in Fig. 5. In fig. 5 we have 3-D mesh 4 x 4 x 4 which is labeling using a Hamiltonian path that expressed in Section 3.1 for each layer. The broadcast will be sent to all nodes in 3-D mesh via the source node which have Hamilton labeled 9 at layer 2 and have an integer coordinate (1, 1, 1)

The routing pattern of Y-HLB algorithm is shown with bold lines as shown in Fig. 5 for each layer, where first layer have y coordinate =0, second layer have y coordinate =1, third layer have y coordinate =2 and fourth layer have y coordinate =3.

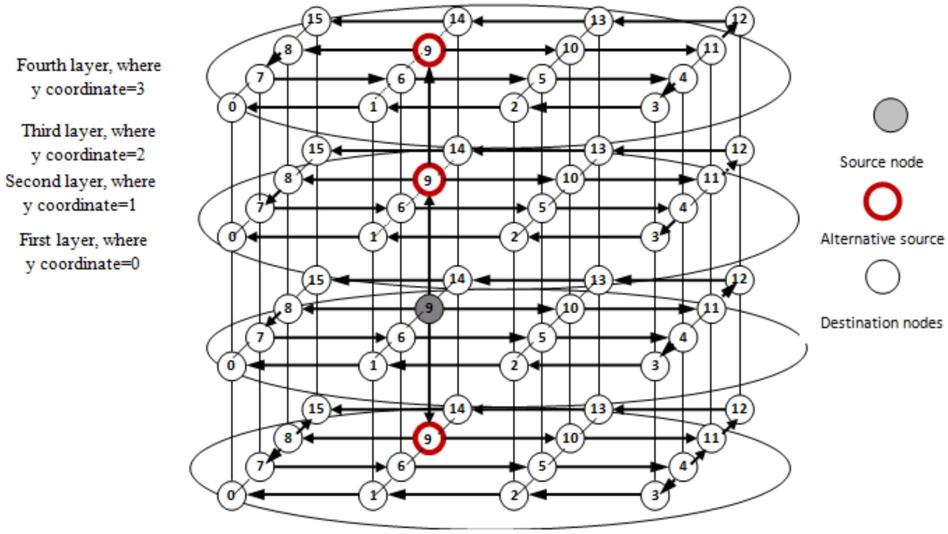

Fig. 5 The routing patterns of Y-HLB -DSPHB algorithm

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 [36]. One of the most known simulations that is used to execute testing in parallel way and provides a very realistic environment is known as CSIM. The simulation program for broadcast communication is part of a larger simulator, called MultiSim [37]. To simulate broadcast routing in 3-D mesh with different sizes, we can use MultiSim.

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 Y-HLB, we compare its performance with 3-DHB and 3-DSPHB algorithms that is presented in [33]. 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. 6. The advantage of the Y-HLB algorithm is significant. In Y-HLB the source node divided the net into n subnets (no. of columns) as shown in section 3.2 and each subnets works as independent layer, so the path from source node to destination nodes in each layer 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 Y-HLB algorithm is less sensitive as the length of the message as shown in Figure 6. 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.

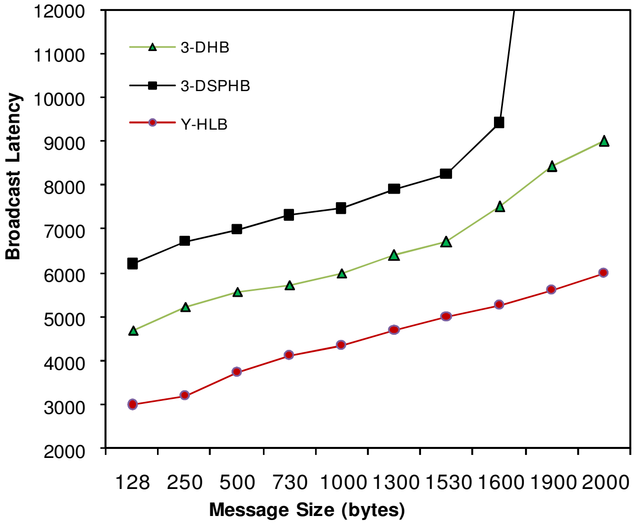

Fig. 6 Performance Comparison of Y-HLB algorithm with 3-DHB and 3-DSPHB algorithms under different message size

Figs 7 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 Y-HLB is the best. Fig. 7 shows us that the disadvantage behavior of the 3-DSPHB algorithm is increases with the injection rate increased.

The source node in Y-HLB 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 y-HLB are shortest paths from source to destinations (set of individual layers), its performance dose not effect when loads on net is very high. So y-HLB is the best good performance.

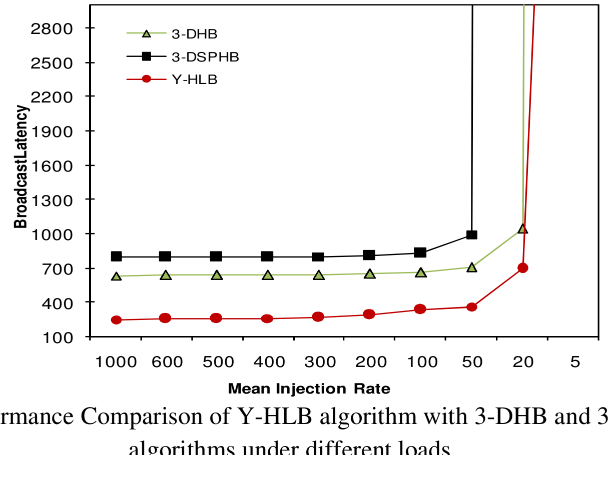

Fig. 7 Performance Comparison of Y-HLB algorithm with 3-DHB and 3-DSPHB algorithms under different loads

5. CONCLUSION

One of the most important parallel communications is broadcast approach, which is highly demanded in parallel applications that are implemented on massively parallel computers.

In this paper we introduce a new algorithm for broadcast wormhole routing in 3-D mesh. We explain our proposed broadcasting algorithm (Y-HLB) which can deliver the message to all nodes in the mesh that connected to the source node. Our proposed algorithm is easy to use and understood. We have also shown that our proposed algorithm is deadlock-free. The performance study shows that the proposed algorithm y-HLB is the best performance than the existing algorithms 3-DHB and 3-DSPHB. Y-HLB 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] 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 & 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] 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] Amnah El-Obaid, (2015), “Three-Dimension Hamiltonian Broadcast Wormhole-Routing”, International Journal of Computer Networks & Communications (IJCNC), Vol.7, No.3.

[34] Amnah El-Obaid and Wan-Li Zuo (2008), “An Efficient Path-Based Multicast Algorithm for Minimum Communication”, Information Technology Journal, 7: 32-39. DOI: 10.3923/itj.2008.32.39

[35] 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

[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. Nehvorks (MASCOTS 93), pp 57-62.

Authors

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