http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 7 (1): 32-39, 2008
ISSN 1812-5638

An Efficient Path-Based Multicast Algorithm for Minimum Communication Steps

Amnah El-Obaid and Wan-Li Zuo
College of Computer Science and Technology, Jilin University, Changchun, China

Abstract: Multicasting is an information dissemination problem which consists, for a processor of a distributed memory parallel computer, to send a same message to a subset of processors. This study presents a new efficient multicast path-based algorithm Two-Path-Pipelined (TPP for short), which can achieve a high degree of parallelism and low communication latency over a wide range of traffic loads in the mesh. Furthermore, its performance is insensitive to the network size, i.e., only two message-passing steps are required to implement a multicast operation irrespective of the network size. TPP algorithm is compared with the GTDMPM algorithm; TPP algorithm has proved to be deadlock free. Results from extensive simulations under a variety of working conditions confirm that the TPP algorithm exhibits superior performance characteristics over GTDMPM algorithm.

Key words: Multicasting communication, wormhole routing, multicomputers, 3-D mesh topology, deadlock-free algorithm

INTRODUCTION

Optimizing the performance of message-passing multicomputers requires matching inter-processor communication algorithms and application characteristics to a suitable underlying interconnection network. The mesh has been one of the most common networks for existing multicomputers due to its desirable properties, such as scalability, ease of implementation, recursive structure and ability to exploit communication locality found in many parallel applications to reduce message latency. Recent interest in multicomputer systems is therefore concentrated on two or three-dimensional mesh and torus networks. Such technology has been adopted by the Intel Touchstone DELTA (Intel Corporation, 1990), MIT J-machine (Nuth and Dally, 1992), Intel Paragon (Foschia *et al*., 1997; Almasi and Gottlieb, 1994), Caltech MOSAIC (Athas and Seitz, 1988) and Cray T3D and T3E (Lessler and Schwazmeier, 1993; Cray Research Inc, 1995).

The switching method determines the way messages visit intermediate nodes. Wormhole switching has been widely used in practice due firstly to its low buffering requirements, allowing for efficient router implementation. Secondly and more importantly, it makes latency almost independent of the message distance in the absence of blocking (Al-Dubai *et al*., 2006). In Wormhole-routed networks, packets are divided into flits. A flit is the smallest unit of information that a channel can accept or refuse. Wormhole routing operates by advancing the head of a packet directly from incoming to outgoing channels (Dally and Seitz, 1987). The transmission from the source node to the destination node is done through a sequence of routers. All flits in the same packet are transmitted in order as pipelined fashion. Only the header flit knows where the packet is going and the remaining data flits must follow the header flit. Once the header flit gains access to a channel, the current message owns that channel until the tail flit passes through it and resigns ownership of the channel. If the header encounters a channel already in use, it is blocked until the channel is freed.

An important primitive among collective communication operations is multicast communication. Multicast is defined as sending a single message from a source node to a set of destination nodes. The performance of multicast communication is measured in terms of its latency in delivering a message to all destinations. Multicast latency consists of three parts, start-up latency network latency and blocking latency (McKinley *et al*., 1995; Duato *et al*., 2003; McKinley *et al*., 1994; Panda *et al*., 1999; Fleury and Fraigniaud, 1998; Malumbres and Duato, 2000; Tseng *et al*., 1996; Chin *et al*., 2000). 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 delays, 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, while blocking latency accounts for delays due to message contention over network resources, e.g., buffers and channels.

In wormhole routing, contiguous flits in a packet are always contained in the same or adjacent nodes of the network. This can cause difficulties, as possibility of

Corresponding Author: Amnah El-Obaid, College of Computer Science and Technology, Jilin University, Changchun, China

deadlock arises. Deadlock in the interconnection network occurs when a set of messages is blocked forever because each message in the set holds one or more resources needed by another message in this set (Hwang, 1993). No communication can occur over the deadlocked channels until exceptional action is taken to break the deadlock. Many deadlock-free routing algorithms have been developed for wormhole communications networks (Malumbres and Duato, 2000; Chin *et al*., 2000; Lin *et al*., 1994; Dianne *et al*., 2001; Jianxi Fan, 2002; Moharam *et al*., 2000; Darwish *et al*., 2005; Al-Dubai *et al*., 2006; Al-Dubai and Ould-Khaoua, 2003).

This study proposes the Two-Path-Pipelined Routing (or TPP for short) as a new routing approach that will be used to devise a new efficient multicast algorithm for the 3-D mesh. Owing to the properties of the TPP, the proposed multicast algorithm requires a fixed number of message-passing steps to implement a multicast operation, irrespective of the system size, considerably reducing the effects of the start-up latency. Results from an extensive comparative analysis presented below will reveal that the new multicast algorithm exhibits superior performance characteristics over the well-known GTDMPM algorithm of Refs (Amnah and Wan, 2007).

THE SYSTEM MODEL

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

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

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

The mesh topology is asymmetric due to the absence of the wrap-around connections along each dimension. As a result, nodes may not be connected to the same number of neighbours; those at the corners, edges and middle of the network have four and six neighbours, respectively. In this system, the node consists of a Processing Element (PE) and 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. Messages generated by the processing element are injected into the network through the injection channel. This study considers the Multi-Port router model where routers are able to relay multiple messages simultaneously provided that each incoming message requires a unique outgoing channel leading to a neighboring node.

THE PROPOSED ALGORITHM (TPP)

This section introduces the Two-Path-Pipelined (TPP for short) multicast algorithm for Multiple-Port 3-D mesh based on the Hamiltonian approach. The proposed algorithm exploits the features of Hamiltonian paths to implement multicast in two message-passing steps, thus considerably reducing the effects of both network size and start-up latency. It splits the destination set into two disjoint subsets (D_U and D_L) and multicasting the message to theses different sets in a pipeline fashion.

A network partitioning strategy based on Hamiltonian paths is fundamental to the deadlock-free routing schemes. A Hamiltonian path visits every node in a graph exactly once; a 3-D mesh has many Hamiltonian paths. In this algorithm, each node u in a multicomputer is assigned a label, L (u). In a network with N nodes, TPP assigns a label for each node based on the position of that node in a Hamiltonian path, where the first node in the path is labeled 0 and the last node in the path is labeled N-1. Fig. 1a shows such a labeling in a 3×3×3 mesh, in which each node is represented by its integer coordinate (x, y, z). The labeling effectively divides the network into two subnetworks. The high-channel subnetwork contains all of the channels whose direction is from lower-labeled nodes to higher-labeled nodes as shown in Fig. 1b and the low-channel network contains all of the channels whose direction is from higher-labeled nodes to lower-labeled nodes as shown in Fig. 1c.

The label assignment function L for an m x n x r mesh can be expressed in terms of the x-, y- and z-coordinates of nodes as follows:

$$\begin{aligned} &\text{If y is even} \\ &L(x,y,z)=\{n*r*y+n*z+x \quad \text{if z is even} \\ &\qquad n*r*y+n*z+(n-x-1)\} \quad \text{if z is odd} \\ &\text{If y is odd} \\ &L(x,y,z)=\{n*r*y+n*(r-z-1)+(n-x-1) \text{ if z is even} \\ &\qquad n*r*y+n*(r-z-1)+x\} \quad \text{if z is odd} \end{aligned} \quad (2)$$

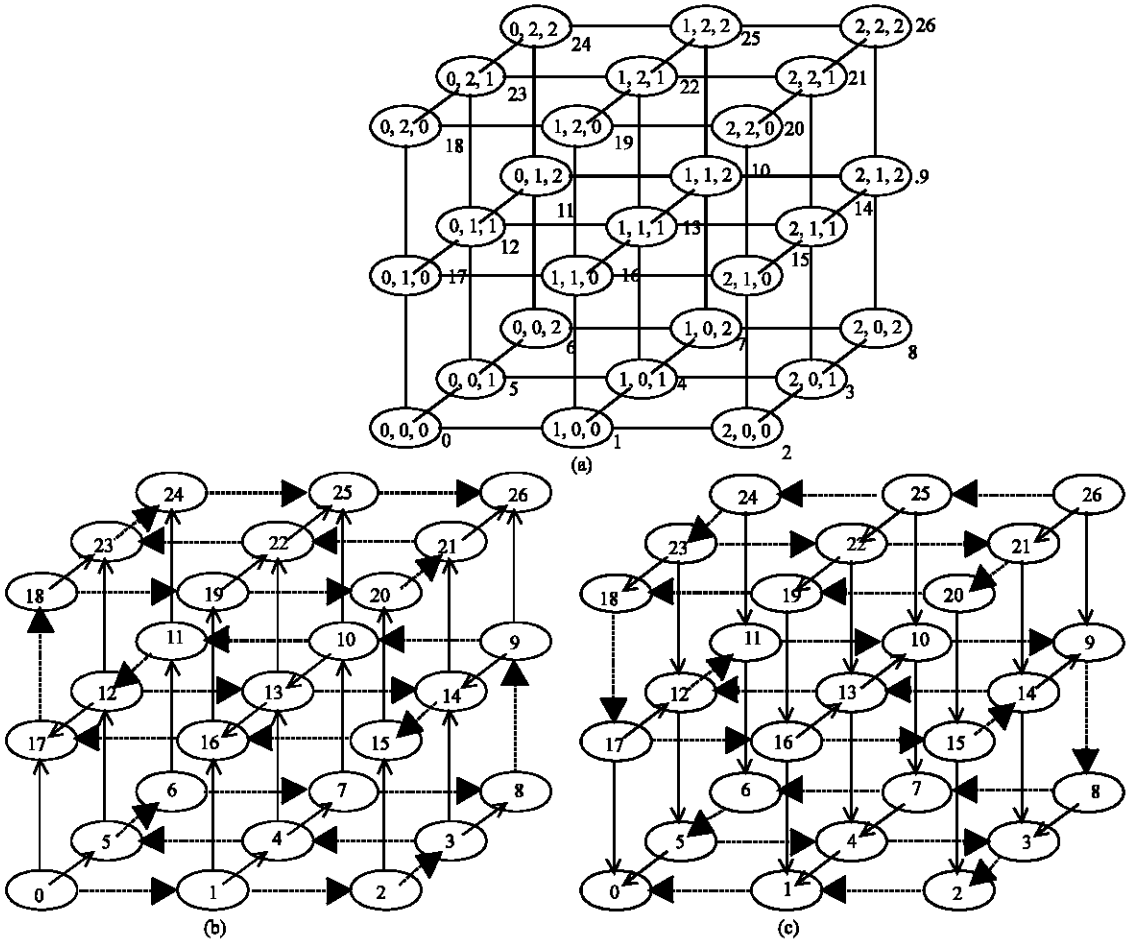

Fig. 1: The labeling of a 3×3×3 mesh, (a) physical network, (b) high-channel network and (c) Low-channel network

At the source node, TPP algorithm divides the network into two subnetworks, N_U and N_L, where every node in N_U has a higher label than that of the source node and every node in N_L has a lower label than that of the source node. The message transmission in TPP is made according to the equation that presented in (Lin *et al*., 1994). One such routing function, defined for a source node u and destination node v, is defined as R (u, v) = w, such that w is a neighboring node of u and, if L (u) < L (v), then we have the following equation:

$$L(w) = \max\left\{ L(z) \,\middle|\, \begin{array}{l} L(z) \le L(v) \text{ and } z \text{ is a} \\ \text{neighboring node of } u \end{array} \right\}$$

or, if $L(u) > L(v)$, then we have the following equation: (3)

$$L(w) = \min\left\{ L(z) \,\middle|\, \begin{array}{l} L(z) \ge L(v) \text{ and } z \text{ is a} \\ \text{neighboring node of } u \end{array} \right\}$$

The simple idea of this algorithm is as follow: -

Step 1: In TPP algorithm, a source node divides the destination set D into two subsets, D_U and D_L, where D_U contain the destination nodes in N_U and D_L contain the destination nodes in N_L. The messages will be sent from the source node to the nodes in D_U using the high-channel network (N_U) and to the destination nodes in D_L using the low-channel network (N_L).

Step 2: Sort the destination nodes in D_U, using the L value as the key, in ascending order. Sort the destination nodes in D_L, using the L value as the key, in descending order.

Step 3: Construct two messages, one containing D_U as part of the header and the other containing D_L as part of the header. The source sends two messages into tow disjoint subnetworks N_U and N_L simultaneously.

Step 4: The TPP routing algorithm uses a distributed routing method in which the routing decision is made at each intermediate node. Upon receiving the message, each intermediate node determines whether its address

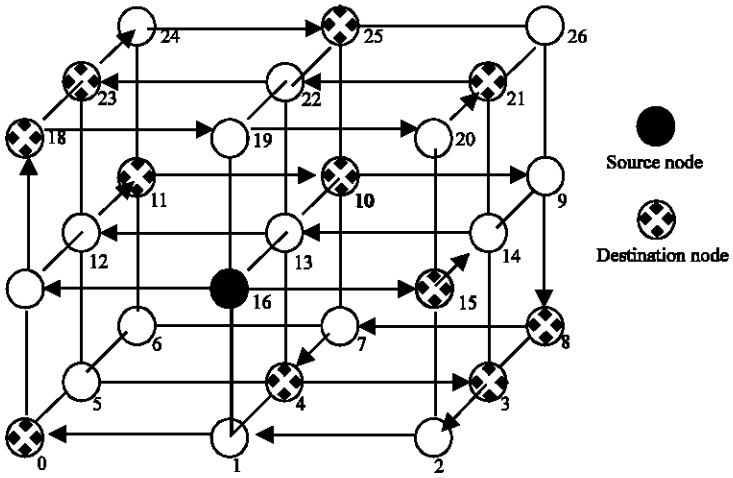

Fig. 2: The routing pattern of GTDTPM

matches that of the first destination node in the message header. If so the address is removed from the message header, the message is copied and sent together with its header to the above (below) neighboring using the routing function R. In case where the intermediate node is not a destination, it sends the message together with its header to the above (below) neighboring using the routing function R.

Step 5: If the sets of the destination nodes are not empty, the algorithm continues according to the previous method.

Consider the example shown in Fig. 2 for a 3×3×3 mesh topology labeling using a Hamiltonian path. The source node labeled 16 initiates a multicast to the destination set D = {0, 4, 3, 8, 15, 10, 11, 18, 21, 23, 25}. The TPP algorithm splits and sorts, D in two subsets D_U = {18, 21, 23, 25} and D_L = {15, 11, 10, 8, 4, 3, 0}. The total number of channels used to deliver the message is 23 (9 in the high-channel network and 14 in the low-channel network). The maximum distance from the source to a destination is 14 hops. The routing pattern is shown with bold lines in Fig. 2.

Assertion 1: TPP is deadlock-free.

Proof: At the source node, TPP algorithm divides the network into two disjoint subnetworks. This is obvious since, $N_U \cap N_L = \varphi$. Then TPP algorithm is deadlock-free at the two subnetworks. Now, we will prove that there are no dependencies within each subnetwork. Since each copy of the message is routed entirely within a single subnetwork and monotonic order (ascending order in N_U and descending order in N_L) of requested channels is guaranteed, there cannot exist a cycle within any subnetwork; hence, no cyclic dependency can be created among the channels. So TPP is deadlock-free.

SIMULATIONS

To compare the performance of our proposed multicast routing algorithms, the simulation program used to model multicast communication in 3-D mesh networks is written in Visual C++ and uses an event-driven simulation package, CSIM (Schwetman, 1985). CSIM allows multiple processes to execute in a quasiparallel fashion and provides a very convenient interface for writing modular simulation programs. The simulation program for multicast communication is part of a larger simulator, called MultiSim (McKinley and Trefftz, 1993), which is designed to study large-scale multiprocessors. MultiSim consists of several components, all of which run within the CSIM package. This section describes the program and results obtained from it. All simulations were executed until the confidence interval was smaller than 5% of the mean, using 95% confidence intervals, which are not shown in the figures. To compare the performance of TPP and GTDMPM, 3-D mesh network that contained single channels is used. We have studied two different values for β. In a first, we have considered β = 10. In a second we have considered β = 100, where β denotes the ratio of the startup time over the propagation time of a flit from one router to a neighboring router. Also we have studied four-message lengths 1, 100, 1000 and 10000 flits. We have studied two sections of experiments, first section study the effects of average injection rate (interarrival time) and second section study the effects of average destination numbers. Two sections are explained below.

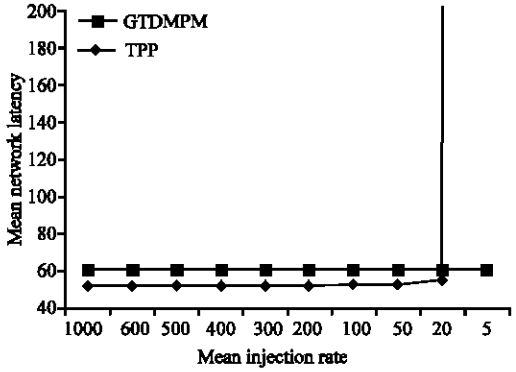

Fig. 3: Performance under different loads. β = 10, message length = 1 flit and No. of destinations = 12

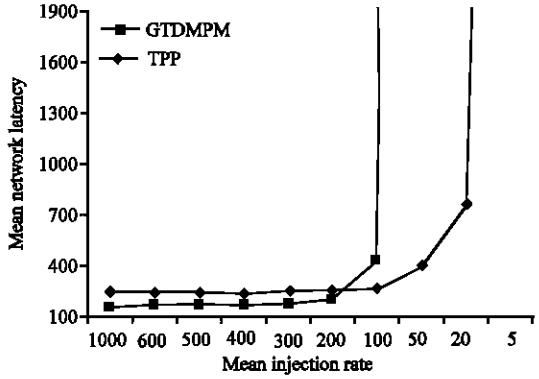

Fig. 4: Performance under different loads. β = 10, message length = 100 flits and No. of destinations = 12

First section (Latency as a function of the injection rate): We have first run simulations on a 5×5×5 mesh. The aim of this first set of experiments is to study the effects of average injection rate (average interarrival time) on our proposed algorithms. For our first set of simulations, we have fixed the number of destination nodes as 10% of the total number of nodes of the mesh, we have studied four message lengths 1, 100, 1000 and 10000 flits and we have studied two different start-up times (β = 10, β = 100).

Figure 3 gives the plot of average network latency for various network loads. The average number of destinations for a multicast is 12, the message length is 1 flit and β is 10. Two algorithms exhibit good performance at low load. Because message size is very small and start-up time is small (1 flit, β = 10), there is no contention in the network due to other multicasts, so two algorithms exhibit good same performance without effect the loads. TPP algorithm exhibits lower latency than GTDMPM algorithm; this result will be explained shortly. The TPP algorithm saturates when load is smaller than 20.

Figure 4 compares two algorithms, again. The message length is 100 flits. The other parameters are the same as for the previous figure. GTDMPM algorithm obtains slight improvement over TTP when the load is greater than 100. This is likely due to the fact that GTDMPM routing introduces less traffic to the network. Then the TTP algorithm turns to be more efficient for a small number of loads, the contention in network due to other multicasts is limited, so the performance depends strongly on the start-up time that required for each algorithm. Because GTDMPM algorithm all-ports, it takes p start-up times (p is the number of destination subsets), while TPP takes only two start-up times. So the performance of TPP is better than GTDMPM at low loads. The TPP algorithm saturates when load is smaller than 20 and GTDMPM algorithm saturates when load is smaller than 200.

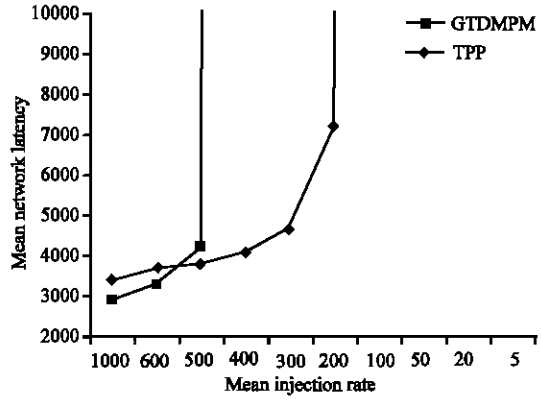

Fig. 5: Performance under different loads. β = 10, message length = 1000 flits and No. of destinations = 12

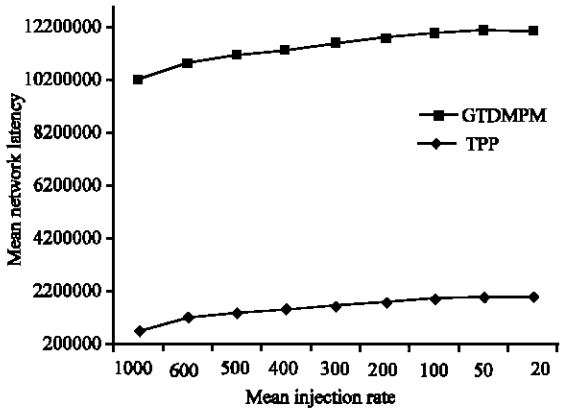

Fig. 6: Performance under different loads. β = 10, message length = 10000 flits and No. of destinations = 12

The TPP algorithm, however, is less sensitive to increased load than the GTDMPM algorithm. The disadvantage of GTDMPM algorithm increases with the message lengths as shown in Fig. 5 and 6.

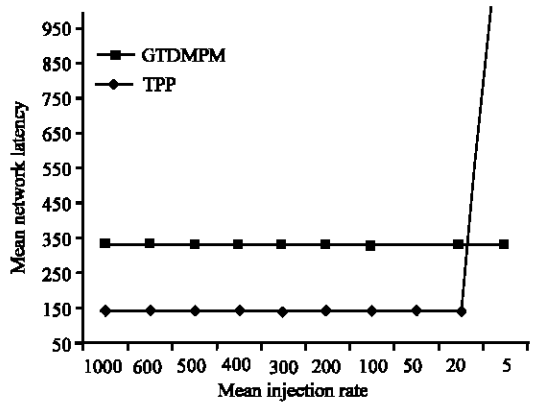

Fig. 7: Performance under different loads. β = 100, message length = 1 flit and No. of destinations = 12

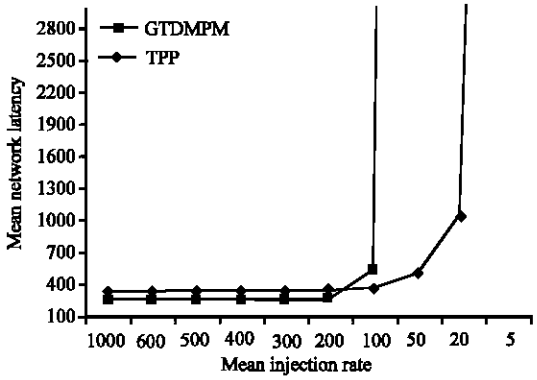

Fig. 8: Performance under different loads. β = 100, message length = 100 flits and No. of destinations = 12

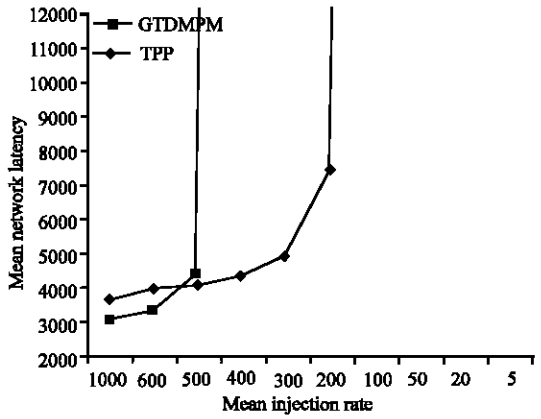

Fig. 9: Performance under different loads. β = 100, message length = 1000 flits and No. of destinations = 12

Figure 7-10 give the plot of average network latency for various network loads. The β is 100; the other

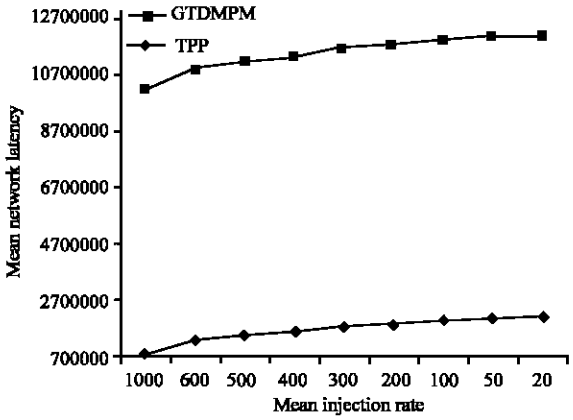

Fig. 10: Performance under different loads. β = 100, message length = 10000 flits and No. of destinations = 12

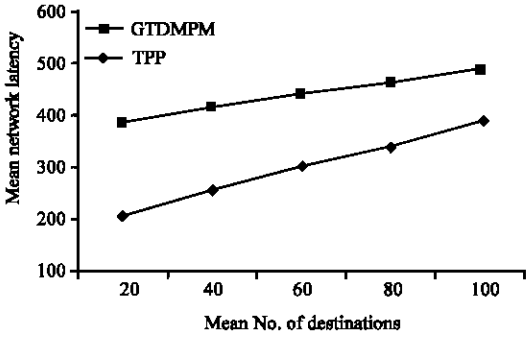

Fig. 11: Performance of different number of destinations. β = 10, message length = 100 flits and mean interarrival time = 300 μ sec

parameters are the same as for the previous figures. The shapes of the curves are similar to shapes of Fig. 3-6, they have same analyze as the one for small start-up time (β = 10).

Second section (latency as a function of the number of destinations): We have run simulations on a 5×5×5 mesh. The aim of this second set of experiments is to study the effects of average destination nodes on our proposed algorithms. In this set of tests, every node generates multicast messages with an average time between messages of 300 Fs. We have studied two message lengths 100 flits, 1000 flits and we have studied two different values for β = (10, 100)

Figure 11 show the network latency obtained by the two algorithms versus various values of number of destinations, ranging from 20 to 100. In this set of tests, every node generates multicast messages with an average time between messages of 300 Fs; the message length is

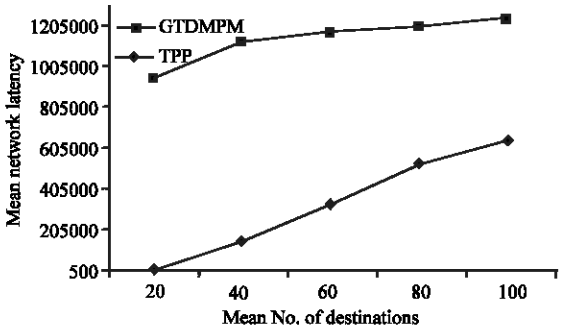

Fig. 12: Performance of different number of destinations. β = 10, Message length = 1000 Flits and Mean interarrival time = 300 μ sec

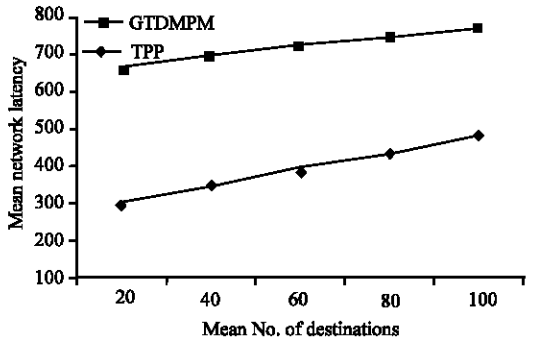

Fig. 13: Performance of different number of destinations. β = 100, Message length = 100 Flits and Mean interarrival time = 300 μ sec

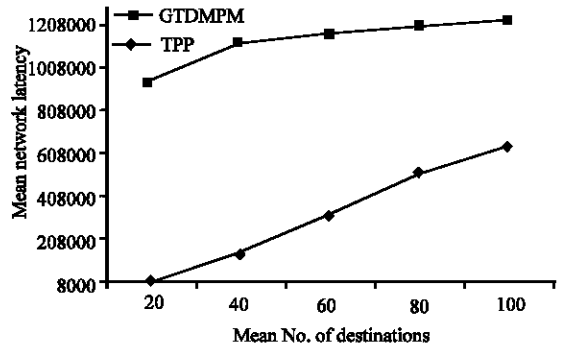

Fig. 14: Performance of different number of destinations. β = 100, message length = 1000 flits and mean interarrival time = 300 μ sec

100 flits and small start-up time (β = 10). Notice that the TPP algorithm results in lower latency than the GTDMPM algorithm for large destination, the reason is somewhat subtle. When GTDMPM algorithm is used to reach a large set of destinations, the source node will send on all of its outgoing channels. Until this multicast transmission is complete, any flit from another multicast message that routes through that source node will be blocked at that point. The source node becomes a "hot point". In fact, every node currently sending a multicast message is likely to be a hot point. If the load is very high, these hot points may decrease system throughput and increase message latency. While in TPP algorithm the source node will send on only two of its outgoing channels, hot points are less likely to occur, the behavior of TPP algorithm is stable under high loads with large destination sets. However, the disadvantage of GTDMPM algorithm increases with the message lengths.

Figure 12 compares two algorithms, again. The message length is 1000 flits. The other parameters are the same as for the previous figure. TPP algorithm exhibits lower latency than GTDMPM algorithm.

Figure 13 and 14 give the plot of average network latency for various values of number of destinations. The β is 100; the other parameters are the same as for the previous figures. The shapes of the curves are similar to shapes of Fig. 11 and 12, they have same analyze as the one for small start-up time (β = 10).

CONCLUSION

In this study, a new deadlock-free multicast wormhole algorithm in 3-D mesh parallel machines using a path-based facility was presented. The proposed algorithm TPP has the main advantage of exhibiting a high degree of parallelism and requiring only two message-passing steps irrespective of the network size. Furthermore, a performance analysis has revealed the best overall performance over well known the GTDMPM algorithm. The disadvantage of GTDMPM algorithm is that hot spots may occur under certain conditions, potentially degrading communication performance.

REFERENCES

Al-Dubai, A.Y. and M. Ould-Khaoua, 2003. A new scalable broadcast algorithm for multiport meshes with minimum communication steps. Microprocessors Microsyst., 27: 101-113.

Al-Dubai, A.Y., M. Ould-Khaoua and L. Mackenzie, 2006. On balancing network traffic in path-based multicast communication. Future Generation Comput. Syst., 22: 805-811.

Almasi, G.S. and A. Gottlieb, 1994. Highly Parallel Computing Benjamin/Cummings.

Amnah, El-Obaid and Wan Li-Zuo, 2007. Deadlock-free multicast wormhole algorithm in 3-D mesh multicomputers. Inform. Technol. J., 6 (5): 623-632.

Athas, W.C. and C.L. Seitz, 1988. Multicomputers: Message passing concurrent computers. IEEE Comput., 21 (8): 9-24.

Chin, T.S., C.Y. Chang and J.P. Sheu, 2000. Efficient path-based multicast in wormhole-routed mesh networks. J. Syst. Architecture, 46: 919-930.

Cray Research Inc, 1995. CRAY T3E scalable parallel processing system. Cray Research Inc., http://www.cray.com/products/systems/crayt3e/.

Dally, W.J. and C.L. Seitz, 1987. Deadlock-free message routing in multiprocessor interconnection networks. IEEE Trans. Co. C-36 (5).

Darwish, M.G., A.A. Radwan, M. Abd El-Baky and K. Hamed, 2005. GTTPM B AN Efficient deadlock-free multicast wormhole algorithm for communication in 2D torus multicomputers. IJICIS, 5 (1).

Dianne, R. Kumar, Walid A. Najjar and Pradip K. Srimani, 2001. A new adaptive hardware tree-based multicast routing in k-ary n-cubes. IEEE Trans. Comput., 50 (7): 647-659.

Duato, J., C. Yalamanchili and L. Ni, 2003. Interconnection Networks: An Engineering Approach. Elsevier Science.

Fleury, E. and P. Fraigniaud, 1998. Strategies for path-based multicasting in wormhole-routed meshes. J. Parallel Distrib. Comput., 60: 26-62.

Foschia, R., 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 Comput. Soc. Press, pp: 117-124.

Hwang, K., 1993. Advanced Computer Architecture: Parallelism, Scalability, Programmability, McGraw-Hill, New York.

Intel Corporation, 1990. A Touchstone DELTA System Description. Intel Corporation. Intel Supercomputing Systems Division.

Jianxi Fan, 2002. Hamilton-connectivity and cycle-embedding of the Mobius cubes, Inform. Processing Lett., 82 (2): 113-117, 30.

Lessler, R.E. and J.L. Schwazmeier, 1993. Cray T3D: A new dimension for cray research, in compcon. IEEE Comput. Soc. Press, pp: 176-182.

Lin, X., P.K. McKinley and L.M. Ni, 1994. Deadlock-free multicast wormhole routing in 2-D mesh multicomputers. IEEE Trans. Parallel Distrib. Syst., 5 (8): 793-804.

Malumbres, M.P. and J. Duato, 2000. An efficient implementation of tree-based multicast routing for distributed shared-memory multiprocessors. J. Syst. Architecture, 46: 1019-1032.

McKinley, P.K. and C. Trefftz, 1993. MultiSim: A tool for the study of large-scale multiprocessors. In Proceeding 1993 International Workshop on Modeling, Analysis and Simulation of Comput. Telecommun. Nehvorks (Mascots), pp: 57-62.

McKinley, P.K., H. Gu, A. Esfahanian and L.M. Ni, 1994. Unicast-based multicast communication in wormhole-routed direct networks. IEEE TPDS, 5 (12): 1254-1265.

McKinley, P., Y.J. Tsai and D. Robinson, 1995. Collective communication in wormhole-routed massively parallel computers. IEEE Comput., 28 (12): 39-50.

Moharam, H., Abd M. El-Baky and S.M.M., 2000. Yomna-An efficient deadlock-free multicast wormhole algorithm in 2-D mesh multicomputers. J. Syst. Architecture, 46 (12): 1073-1091.

Nuth, P.R. and W.J. Dally, 1992. The J-machine network. In Proc. IEEE Int. Conf. on Computer Design: VLSI in Computer and Processors. IEEE Comput. Soc. Press, pp: 420-423.

Panda, D.K., S. Singal and R. Kesavan, 1999. Multidestination message-passing in wormhole k-ary n-cube networks with base routing conformed paths. IEEE TPDS, 10 (1): 76-96.

Schwetman, H.D., 1985. CSIM: A C-based, process-oriented simulation language. Tech. Rep. Microelectronics Comput. Technol. Corp, pp: 80-85.

Tseng, Y., D.K. Panda and T. Lai, 1996. A trip-based multicasting model in wormhole-routed networks with virtual channels. IEEE Trans. Parallel Distrib. Syst., 7 (2): 138-150.