On the Performance of Broadcast Algorithms in Interconnection Networks

Ahmed Yassin Al-Dubai* and Mohamed Ould-Khaoua**

*Institute for Information Technology, Thames Valley University
Wellington Street, Slough, SL1 1YG, UK
Email: Ahmed.Al-Dubai@tvu.ac.uk

**Department of Computing Science, University of Glasgow
Glasgow, G12 8RZ
Email: mohamed@dcs.gla.ac.uk

Abstract

Broadcast Communication is among the most primitive collective capabilities of any message passing network. Broadcast algorithms for the mesh have been widely reported in the literature. However, most existing algorithms have been studied within limited conditions, such as light traffic load and fixed network sizes. In other words, most of these algorithms have not been studied at different Quality of Service (QoS) levels. In contrast, this study examines the broadcast operation, taking into account the scalability, parallelism, a wide range of traffic loads through the propagation of broadcast messages. To the best of our knowledge, this study is the first to consider the issue of broadcast latency at both the network and node levels across different traffic loads. Results are shown from a comparative analysis confirming that the coded-path based broadcast algorithms exhibit superior performance characteristics over some existing algorithms.

1. Introduction

Mesh interconnection network often stands out as a natural choice for the problem representation in parallel and distributed computing [1-9, 16, 20, 22]. This is mainly due to its desirable properties, such as ease of implementation, scalability, low cross-section bandwidth, and ability to exploit communication locality found in many parallel applications to reduce message latency. In addition, meshes partition into units that are still meshes, simplifying the design of routing algorithms that are independent of network size [5, 20]. Processors (or nodes) of a mesh communicate with each other by sending messages through the underlying network. Regardless of how well the computation is distributed among the processors, communication overhead can severely limit speedup [11]. Hence, efficient communication is critical to the performance of a mesh network. Currently, the most popular technique for switching packets is wormhole switching, in which a message is divided into flow control digits (or flits) [5, 11, 16, 25]. The flits are routed through the network one after another in a pipeline fashion.

Broadcast communication refers to the delivery of the same message, originating from a given source, to all the nodes in the network. This type of communication has for long been among the most important topics in the field of parallel and distributed computing due to its increasing importance in many real-world parallel applications found in the areas of science and engineering [21, 26]. For instance, broadcast is often required in scientific computations to distribute large data arrays over system nodes in order, to perform various data manipulation operations. It is also required in control operations, such as global synchronisation, and to signal changes in network conditions, e.g., faults. In the distributed shared-memory paradigm, broadcast communication is often used to support shared data invalidation and updating procedures required for cache coherence protocols [5]. Intuitively, a practical broadcast algorithm must be deadlock free and capable of broadcasting in few message-passing steps. A number of broadcast algorithms have been proposed for the mesh in the literature [1, 2, 3, 11, 16, 22, 26]. However, most of these algorithms implement broadcast with a significant communication overhead due to the high number of message passing steps (or start up latencies) that a message encounters while propagating inside the

1530-2016/05 $20.00 © 2005 IEEE

network and do not exhibit good scalability properties as the network size increases. As a consequence, most existing broadcast algorithms cannot efficiently support real-world parallel applications that require large-scale system sizes due to their high computational demands. Furthermore most existing broadcast algorithms have focused on the broadcast latency at the network level only, with a little consideration for the variation in message arrival times at the node level, resulting in an erratic variation of the message arrival times at the destination nodes. Motivated by this observation, we have previously proposed deterministic and adaptive broadcast algorithms, which are based on the coded-path routing [1].

The CPR exploits the main features of wormhole switching, such as few buffer requirements and distance insensitivity, to overcome the limitations of the existing approaches, and to efficiently support collective communications. In the CPR, the header flit has two bits that form the control field. The two bits indicate to a router which action to take, e.g., pass or receive, upon the reception of a message. As in the path-based algorithms of [4, 10], which use the multidestination approach, it is assumed that a router in the CPR can simultaneously receive a message and passes a copy to the next router. Due to space limitation, we refer the reader to [1] for more detail on the CPR.

Like the previous studies, these suggested algorithms have been studied previously only under certain conditions, such as light traffic load and fixed network sizes. In other words, these algorithms have not been studied at different Quality of Service (QoS) levels. In contrast, this paper presents a comparison study between broadcast algorithms, taking into account the scalability, parallelism, a wide range of traffic loads and low latency. To the best of our knowledge, this study is the first to consider the issue of broadcast latency at both the network and node levels. Results are shown from a comparative analysis confirming that the coded-path based broadcast algorithms exhibits superior performance characteristics over those of the well-known Recursive Doubling, Extending Dominating Node. The remainder of this paper is organised as follows. Section 2 describes the broadcast algorithms. Section 3 presents our results. Finally, Section 4 provides a summary of this paper.

2. The broadcast algorithms

Most current practical parallel machines, including meshes, employ Dimension-ordered routing, where messages visit network dimensions in a pre-defined order, [2, 5, 20]. This is due to the fact that this form of routing is simple and deadlock free, resulting in fast and compact routers when the algorithm is implemented in hardware, [5]. To exploit these features, many researchers have concentrated on trying to enhance broadcast algorithms by observing that the communication latency is primarily generated by the number of message-passing steps between the involved processors [2, 11, 20, 21, 22]. For instance, Barnett *et. al.* [2] have originally proposed the Recursive Doubling (RD) algorithm. This algorithm requires $\log_2 N$ message-passing steps for broadcasting in an N-node mesh. In each message-passing step, each node holding a copy of the message is responsible for a partition of a row or column. The node divides its partition in half and sends a copy of the message to the node in the other half that occupies the same relative position. This process is implemented recursively until the completion of the broadcast operation. In the absence of message contention, the RD algorithm can take advantage of the pipelining feature of wormhole switching [2, 20]. Although the RD algorithm can be implemented on a multi-port router architecture, it is often unable to take advantage of this architecture model, resulting in more message-passing steps being than necessary. In fact, the number of steps is still $\log_2 N$ [20]. To utilise the multi-port facility of multi-ported meshes, the Extended Dominating Nodes algorithm (EDN) has been proposed by Tsai and McKinley [20].

EDN was proposed by Tsai and McKinley [20], and can systematically construct collective operations in multiport wormhole-routed networks. In the EDN approach, the network is divided into several levels. For each level, a dominating set is assigned. For instance, a dominating set D of a graph G is a set of vertices in G such that every vertex in G is either in D or is adjacent to at least one vertex in D. However, EDN requires that the number of nodes along a given dimension be multiple of 4. The authors in [20] have shown that the number of message-passing steps required in a network size of $(4\times2^k)\times(4\times2^k)\times(4\times2^m)$ or $(4\times2^k)\times(4\times2^k)\times(4\times3^m)$ *is* $k+m+4$, where k and $m\geq0$.

In addition, even though EDN can take advantage of a multiport router model (we assume that the EDN uses a three port router in the present study), the number of message our previous proposed algorithm is based on this routing approach. To minimise the effect of the network size by reducing the number of message-passing required, we have previously proposed, in [28], the Deterministic Broadcast (DB for short) algorithm for the *n*-dimensional mesh based on the dimension ordered and Coded Path Routing discussed in *CPR* [1]. The *DB* algorithm exploits the features of *CPR* to implement the broadcast operation in few message-passing steps, thus considerably reducing the effects of start-up latency. For simplicity, here we apply *DB* to the 2D mesh, although our discussion can be easily extended to the higher dimensional mesh. To broadcast with the *DB* algorithm, the mesh topology is divided into two sets of nodes, row partitioning, and column partitioning sets. For each partitioning set, a corner node is selected. The source node initiates the broadcast message and sends it to the two selected corners. Each corner node acts as a source node, disseminating the message to one side of the partitioning set. Upon receiving the message, each selected side sends the message to the opposite side in its partitioning set, covering the rest of the nodes of the system in parallel [28]. While most previous broadcast algorithms, including the *DB* algorithm described above, have been discussed in the context of deterministic routing [2, 11, 20, 21, 22], this section focuses on the Adaptive Broadcast (AB for short) algorithm presented in [27].

The AB algorithm uses *CPR* along with adaptive routing in order to enable broadcast messages to exploit alternative paths existing in the network to cross from source to destination. For the sake of the current discussion, the *AB* algorithm uses the Turn model routing presented in [6] to achieve routing adaptivity while ensuring deadlock freedom [5, 6, 7]. We restrict our discussions here to the Turn model because it is a general model for designing wormhole-based adaptive routing algorithms [5, 6, 7]. However, the *AB* algorithm can be employed with other underlying adaptive routing models as well. While the Turn model routing prohibits just enough turns to ensure deadlock freedom, its adaptivity provides the *AB* algorithm with flexibility in choosing a network path during a given message-passing step. In this study, the west first-turn model routing is used, which prohibits the turns (south, west) and (north, west) as discussed in [6]. The proposed algorithm implements the broadcast operation in only three message passing steps in 2D thus considerably reducing the effects of both network size and start-up latency. When the *AB* algorithm is used in the 3D mesh, the network is treated as a set of planes, each representing a 2D sub-mesh. In *AB* algorithm, from any starting node, a message is routed to the nearest corner of the plane that includes the source node, and also to the opposite corner of that plane. The value of the control field is set at 10. Then, each selected corner, in turn, acts as a source node and sends the message to the corresponding corners in the other planes of the system after changing the control field to 11. Thus, every plane in the system receives a copy of the broadcast message via two corners in parallel. Finally, every plane is divided in half and messages are disseminated from the selected corners, to cover all remaining sub-mesh nodes. The *AB* algorithm uses the strategy of limiting the number of destination nodes for each message path, thereby preventing excessively long total path lengths.

3. Performance evaluation

This section compares the performance of the proposed algorithms, DB and AB, to the well-known Recursive Doubling (RD) [2] and Extended Dominating Node (EDN) algorithms [20] (both based on dimension order routing [5, 10]). For the sake of clarity and brevity in what follows the four broadcast algorithms will be referred to as simply DB, AB, RD, and EDN. In our performance study, simulation experiments are conducted to evaluate the behaviour of the algorithms under different traffic conditions. To analyse the performance of the broadcast algorithms and verify the timing analysis of DB, AB, RD and EDN, a simulator has been developed to model the broadcast operations of the four algorithms in the 3D mesh.

The results have been found not to change much when the other versions of the mesh topology have been considered. The package was written in VC++ and used the MultiSim software [13], built on top of the event-driven CSIM-18 package [18]. In the simulator, processes are used to model the active entities of a system and execute in a quasi-parallel fashion, providing a convenient means for writing a modular simulation program. In our case every node is modelled as a process. For studying broadcast communication, the main program activates a set of CSIM parallel processes that are used to broadcast a message in the network. In each experiment, different source nodes have been chosen randomly using a uniform number generator. Each broadcast message is simulated with a pseudo-process that sends the messages to the destinations by creating path processes.

A path is an alternating sequence of nodes and channels traversed by a message. In our experiments, we have considered a wide range of message lengths, e.g. 32 to 2048 flits. As in the previous studies of [2, 20], two values for the start-up latency have been considered, notably T_s=0.15 and 1.5 μs, and the time required to transmit a flit on a channel $\beta = 0.003\,\mu s$. The values selected for the T_s and β are consistent with the implementation technology used in current systems, such as the Cray T3D machine [8]. In each broadcast algorithm the path processes uses the underlying routing scheme to determine the channels on which each message should be transmitted. Each channel has a single queue where messages are held while awaiting transmission. Statistics have been collected with a 95% confidence interval when the system reaches a steady state (i.e., when results do not change with time).

3.1 Effects of the network size and start-up latency

To evaluate the effects of the network size on the performance of the four algorithms, network sizes between 64 and 4096 nodes have been simulated. In this set of simulations, we consider the case of a single-source broadcast like [2, 20]. Fig. 1 shows that both DB and AB outperform their RD and EDN counterparts. Although EDN and DB have comparable performance for a small network size, notably for the 3D-mesh with $4 \times 4 \times 4$ nodes, the performance of EDN degrades substantially as the network size increases. This is because both DB and EDN require the same number of message-passing steps when $4 \times 4 \times 4$ is considered as a network size. Nonetheless, the larger the network size, the longer the latency incurred by the EDN becomes. In contrast, DB and AB provide a better performance irrespective of the network size since there is no increase in the number of message-passing steps required to complete a broadcast operation in larger networks.

3.2 Latency at the node level

Existing broadcast algorithms have been designed with a consideration paid only to the latency at the network level, resulting in a large variation of the message arrival times at the destination nodes. This study is the first to consider latency at both these levels in the comparative analysis. To this end, the coefficient of variation in message arrival times at the destination nodes in *DB* and *AB* have been compared against that of *EDN* and *RD* algorithms. We have computed the coefficient of variation of the communication latency among the destination nodes. This measure reflects the degree of parallelism achieved by the broadcast algorithms. The message length has been fixed at *L*=64 flits. The network size has been varied from *N*=64 to 1024 nodes. The coefficient of variation, CV, is defined as SD / M_{nl}, where SD refers to the standard deviation of the message arrival times among the destination nodes and M_{nl} is the mean communication latency.

Tables 1 and 2 contain performance results of *DB* and *AB, EDN* and *RD*. The results are obtained by averaging the values from at least 40 experiments. The (DB_{IMR}%) and (AB_{IMR}%) in Tables 1 and 2 refer to the percentage improvement obtained by *DB* and *AB*, respectively. *DB* and *AB* achieve a significant improvement over *EDN* and *RD*. This is due to two main reasons. Firstly, owing to the partitioning scheme adopted by the proposed algorithms, *DB* and *AB*, that divides the destinations into comparable partitions. Secondly, and more importantly, unlike in *RD* and *EDN*, most of the destination nodes in each partition receive the broadcast message within the same message-passing step. This has the net effect of minimising the variance of the arrival times at the node level. In contrast, *RD* and *EDN* implement the broadcast operation with a higher number of message-passing steps, which in turn, leads to substantial variations in the arrival times at the destination nodes. It is worth noting that *DB* and *AB* exhibit comparable performance behaviour, with *DB* having a slightly lower coefficient of variation, especially for small network sizes (e.g. N=64 nodes) as shown in Fig. 2. However, the superiority of *AB* over *DB* diminishes in larger network sizes (e.g. *N*=1024 nodes). This is due to the length of the paths used by the two algorithms. Despite the fact that *AB* requires less message-passing steps, (three are required) than *DB*, (four are required), the former algorithm uses longer paths in its third step, resulting in a higher coefficient variation of the arrival times among the destination nodes.

3.3 Performance in the presence of unicast and broadcast

The above section has considered the case of a single node initiating a broadcast operation in the network. However, it commonly happens in practical applications that a number of nodes send broadcast and unicast messages simultaneously. In such scenarios, the performance behaviour of the four broadcast algorithms may differ from that reported in the above section as contention may occur among messages over networks resources (e.g. buffers and channels). To study the performance of the proposed algorithms in the presence of unicast and broadcast messages, simulation experiments have been conducted, considering the 3D mesh of two sizes, notably $8\times8\times8$ and $16\times16\times8$ nodes. Traffic generated from a given source node contains 90 percent unicast messages and 10 percent broadcast messages. A source node is randomly chosen for a broadcast operation. Nodes generate messages at time intervals chosen from an exponential distribution.

Figs. 3 and 4 report the results of the communication latency as a function of the traffic load. Fig. 3 shows the results for the $8\times8\times8$ mesh. The proposed DB and AB algorithms offer a much better performance for both network throughput and communication latency over EDN and RD. This is because the former algorithms require a minimum number of message passing steps, resulting in a lower traffic load during the broadcast operation. A batch strategy has been used to compute the mean communication latency where 20 batches have been used to collect the statistics reported here (actually 21 batches were used, but the first batch statistics have been ignored because it produces optimistic values due to cold start). However, EDN and RD use a higher number of messages, leading to competition among messages for network resources.

In the face of such competition, nodes may experience congestion, resulting in an early saturation and high communication latencies. The results presented in Fig. 6 show that AB exhibits the best performance followed by DB. This is mainly due to the fact that AB is based on adaptive routing, which provides messages with alternative paths inside the network. However, in a large network of $16\times16\times8$ nodes, this performance advantage diminishes, as depicted in Fig. 4. Although AB requires a lower number of message-passing steps, it uses longer paths to propagate a broadcast message to all network nodes. These long paths increase load inside the network substantially, resulting in a reduction in throughput. The larger the networks size, the lower the performance advantage of *AB* becomes. Nevertheless, this algorithm still offers the best performance in terms of throughput and communication latency under light traffic.

4. Conclusion and future directions

Most existing broadcast algorithms have been studied within limited conditions, such as light traffic load and fixed network sizes. In other words, most of these algorithms have not been studied at different Quality of Service (QoS) levels. In contrast, this paper presents a comparison study between broadcast algorithms, taking into account the scalability, parallelism, and a wide range of traffic loads. Extensive simulation experiments under a variety of traffic conditions have been conducted to compare the performance of the *DB* and *AB* algorithms against the well-known existing *EDN* and *RD* broadcast algorithms. Different network sizes, e.g., 64 to 4096 nodes have been examined in the simulation experiments. For these network sizes, our analysis has shown that the proposed *DB* and *AB* algorithms exhibit better scalability properties than their *EDN* and *RD* competitors. The simulation results have also shown that *DB* and *AB* have a lower broadcast latency than the existing algorithms under various traffic loads. A number of interconnection networks have been proposed for multicomputers over the past years such as the *k*-ary *n*-cube and generalised hypercube. An interesting line of research would be to propose multicast and broadcast algorithms for these common topologies.

5. References

[1] A. Al-Dubai and M. Ould-Khaoua, Coded path routing: A new approach to broadcasting in 3-D meshes, *Proc. 20th IEEE Int. Performance, Computing and Communications Conference(IEEE-IPCCC'2001),* Phoenix, USA, pp 155-162, April 4-6, 2001.

[2] M. Barnett, G. David, R. A. van de Geijn and J. Watts, Broadcasting on meshes with wormhole routing, *J.Parallel & Distributed Computing*, vol. 35, pp. 111-122, 1996.

[3] S. Cang, J. Wu, Time-step optimal broadcasting in 3-D meshes with minimal total communication distance*, J. Parallel & Distributed Computing*, vol. 60, pp. 966-997, 2000.

[4] T.-S. Chin, C.-Y. Chang, J-P. Sheu, Efficient path-based multicast in wormhole-routed mesh networks, *J. Systems Architecture*, vol. 46 , pp. 919-930, 2000.

[5] J. Duato, S. Yalamanchili, L. Ni, Interconnection networks: An engineering approach, *IEEE Computer Society Press*, 1997.

[6] C. J. Glass, L. M. Ni, The Turn model for adaptive routing, *Proc. Ann. Int'l Symp. Computer Architecture*, pp. 278 287, 1992.

[7] Ge-Ming Chiu, The Odd-even turn model for adaptive routing, *IEEE TPPB*, vol. 11, no. 7, pp. 729-738, July, 2000.

[8] R. E. Kessler, J. L. Schwazmeier, CRAY T3D: A new dimension for Cray research, *Proc. COMPCON'93,* pp. 176-182, 1993.

[9] D. Lenoski *et al*., The Stanford DASH multiprocessor, *IEEE Computer*, vol. 25, no. 3, 1992.

[10] X. Lin, L. M. Ni, Deadlock-free multicast wormhole routing multicomputer networks, *Proc. Int. Symp. Computer Architecture*, pp. 116-124, 1991.

[11] P. K. McKinley, Y.-J. Tsai, D. Robinson, Collective communication in wormhole-routed massively parallel computers, *IEEE Computer*, vol. 28, no. 12, pp. 39-50, Dec., 1995.

[12] P. K. McKinley, H. Xu, A. Esfahanian, L. M. Ni, Unicast-based multicast communication in wormhole-routed direct networks, *IEEE TPPB*, vol. 5, no. 12, pp. 1254-1265, 1994.

[13] P. K. McKinley, C. Trefftz, MultiSim: A simulation tool for the study of large-scale multiprocessors, *Proc. MASCOTS'1993*, pp. 57-62, 1993.

[14] L. Ni, P. McKinley, A survey of wormhole routing techniques in direct networks, *Computer*, vol. 26, no. 2, pp. 62- 76, Feb. 1993.

[15] D. K. Panda, Issues in designing efficient and practical algorithms for collective communication on wormhole routed systems, *Proc. 1995 Workshop Challenges for Parallel Processing*, pp. 8- 15, 1995.

[16] D. K. Panda, S. Singal, R. Kesavan, Multidestination message-passing in wormhole *k-ary n*cube networks with base routing conformed paths, *IEEE Trans. Parallel & Distributed Systems*, vol. 10, no. 1, pp. 76-96, 1999.

[17] D. F. Robinson, P. K. McKinley, C. Cheng, Path based multicast communication in wormhole routed unidirectional torus networks, *JPDC*, vol. 45 , pp.104 - 121, 1997.

[18] H. D. Schwetman, CSIM: A C-based, process-oriented simulation language, Tech. Rep. pp. 80- 85, Microelectronics and Computer Technology Corp., 1985.

[19] Y.-J. Tsai, P. K. McKinley, A broadcast algorithm for all-port wormhole-routed torus networks, *IEEE Trans. Parallel & Distributed Systems*, vol. 7, no. 8, pp. 876-895, 1996.

[20] Y.-J Tsai, P.K. McKinley, An extended dominating node approach to broadcast and global combine in multiport wormhole routed mesh networks, *IEEE Trans. Parallel & Distributed Systems*, vol. 8, no. 1, pp. 41 – 58, 1997.

[21] Y.-C. Tseng, S.-Y. Wang, C.-W. Ho, Efficient broadcasting in wormhole-routed multicomputers: A network-partitioning approach, *IEEE Trans. Parallel & Distributed Systems*, vol. 10, no. 1, pp. 44-61, 1999.

[22] J. Watts, Efficient collective communication on multidimensional meshes with wormhole routing. Tech. Rep. TR 94-19. Dept. Computer Science, Univ. Texas at Austin, June 1994.

[23] M. P. Malumbres, J. Duato, An efficient implementation of tree-based multicast routing for distributed shared memory multiprocessors, *J. Systems Architecture*, vol. 46, 1019-1032, 2000.

[24] Lionel M. Ni, Issues in designing truly scalable interconnection networks, *the Workshop of the Int. Conf. on Parallel Processing (ICPP)*, pp 74-83*,* August 12, 1996.

[25] A. A. Chien, A cost and speed model for *k-ary n*-cube wormhole routers, *IEEE Trans. Parallel and Distributed Systems*, vol. 9. no. 2, pp. 150-160, 1998.

[26] S. Dobrev, I. Vrto, Optimal broadcasting in tori with dynamic faults, *Parallel Processing Letters*, vol. 12, no. 1, pp. 17-22, 2002.

[27] A. Y. Al-Dubai , M. Ould-Khaoua and L.M. Mackenzie, A scalable plane-based broadcast algorithm for multicomputers, *Proc. of the 11th Euro-micro Conference on Parallel Distributed and Network based Processing (IEEE-PDP-*2003), Genoa - Italy, pp 149-156, February 5-7, 2003.

[28] A. Al-Dubai and M. Ould-Khaoua, Towards scalable collective communication for multicomputer interconnection networks, *Journal of Information Sciences, Elsevier,* vol. 163, no. 4, pp. 293-306, 2004.

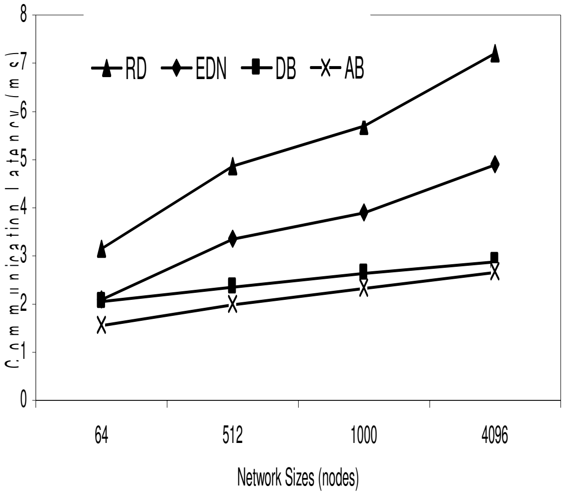

Fig. 1: Communication latency of DB, AB, RD and EDN for various network sizes. Message length *L*=100 flits start-up latency Ts=1.5 μs.

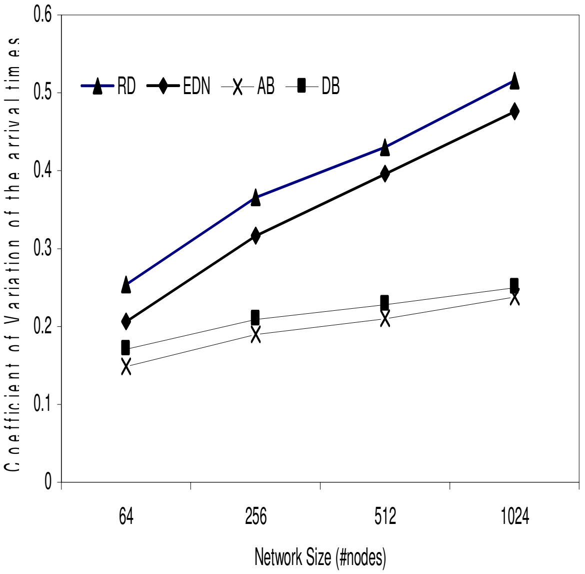

Fig. 2: The coefficient of variation of the DB, AB, RD and EDN algorithms for various network sizes. Message length *L*=100 flits start-up latency Ts=1.5 µs.

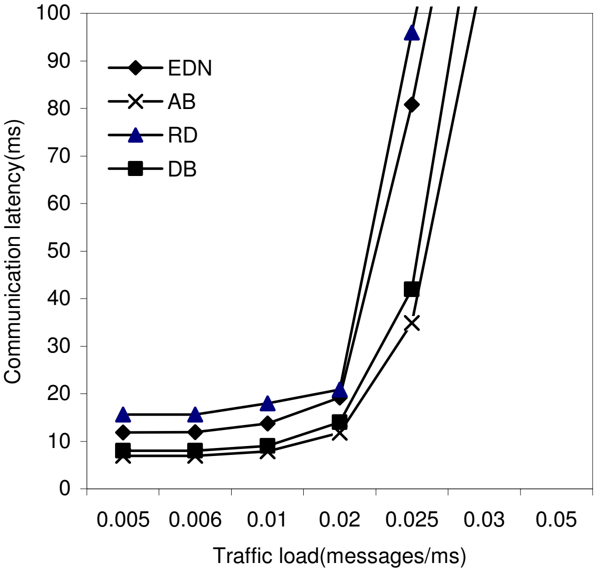

Fig. 3: Communication latency of DB, AB, RD and EDN under different traffic loads. Network size N=$(8\times8\times8)$ nodes, start-up latency Ts=1.5 µs, message length L=32 flits.

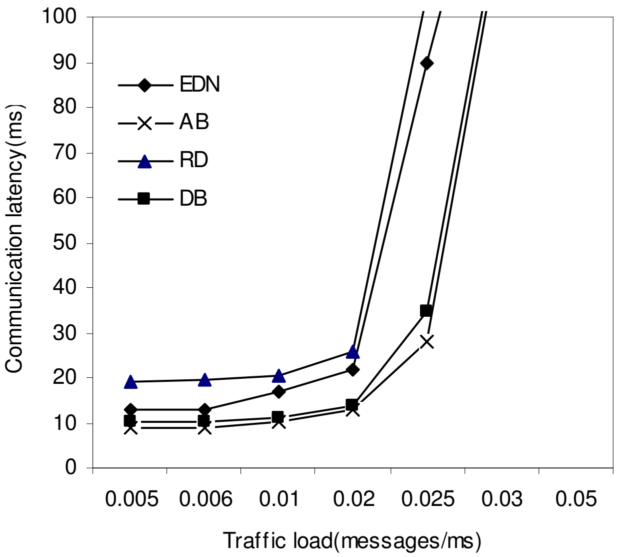

Fig. 4: Communication latency of DB, AB, RD and EDN for various traffic loads. Network size N=$16\times16\times8$ nodes, start-up latency Ts=1.5 µs, message length M=32 flits.

Table 1: Coefficient of Variation (CV) of the broadcast latencies in DB, EDN and RD in the 3D mesh with the improvement obtained by the proposed algorithm (DB$_{\text{IMR}}$%)

Mesh Size	(4×4×4)		4×4×16		8×8×8		8×8×16	
	64 node		256 node		512 node		1024 node	
	CV	$DB_{IMR}\%$	CV	$DB_{IMR}\%$	CV	$DB_{IMR}\%$	CV	$DB_{IMR}\%$
RD	0.2540	65.41	0.3661	84.31	0.4263	92.54	0.5160	109.5
EDN	0.2064	34.32	0.3164	60.34	0.3962	83.33	0.4761	93.34

Table 2: Coefficient of Variation (CV) of the broadcast latencies in AB, EDN and RD in the 3D mesh with the improvement obtained by the proposed algorithm (AB$_{\text{IMP}}$%)

Mesh Size	(4×4×4)		4×4×16		8×8×8		8×8×16	
	64 node		256 node		512 node		1024 node	
	CV	$AB_{IMR}\%$	CV	$AB_{IMR}\%$	CV	$AB_{IMR}\%$	CV	$AB_{IMR}\%$
RD	0.2540	73.844	0.3661	92.87	0.4263	104.65	0.5160	116.81
EDN	0.2064	41.27	0.3164	66.70	0.3962	90.21	0.4761	100.1