Design and Analysis of Multicast Communication in Multidimensional Mesh Networks

Ahmed Al-Dubai[1], Mohamed Ould-Khaoua[2], and Imed Romdhani[3]

[1,3] School of Computing, Napier University, 10 Colinton Road
Edinburgh, EH10 5DT, UK
{a.al-dubai,i.romdhani}@napier.ac.uk
[2] Department of Computing Science, University of Glasgow
Glasgow G12 8RZ, UK
mohamed@dcs.gla.ac.uk

Abstract. This paper addresses the issue of multicast communication in scalable interconnection networks, using path-based scheme. Most existing multicast algorithms either assume a fixed network size, low dimensional networks or only consider the latency at the network level. As a consequence, most of these algorithms implement multicast in a sequential manner and can not scale well with the network dimensions or the number of nodes involved. Furthermore, most of these algorithms handle multicast communication with low throughput. In this paper, we propose a multicast algorithm for multidimensional interconnection networks, which is built upon our Qualified Groups QG multicast scheme for ensuring efficient communication irrespective of the network sizes/dimensions or the number of the destination nodes. Unlike the existing works, this study considers the scalability and latency at both the network and node levels so as to achieve a high degree of parallelism. Our results show that the proposed algorithm considerably improves the multicast message delivery ratio, throughput and scalability.

Keywords: Mesh Networks, Path-based Multicast, Routing Algorithms.

1 Introduction

Multicast is a fundamental communication pattern for ensuring a scalable implementation of a wide variety of applications in parallel and distributed computing. In multicast communication, a source node sends the same message to an arbitrary number of destinations in the network. Multicast is widely used by many important applications. For instance, multicast is frequently used by parallel search and parallel graph algorithms [3, 14]. In more recent fields such as bioinformatics, protein sequences are clustered into families of related sequences. Multicast services are required during the creation and maintenance of theses clusters. Furthermore, multicast communication can be used as a tool that can allow efficient access and update of different types of information and finding the genes in the DNA sequences of various organisms. Moreover, multicast is also fundamental to the implementation

I. Stojmenovic et al. (Eds.): ISPA 2007, LNCS 4742, pp.868–882,2007.

of higher-level communication operations such as gossip, gather, and barrier synchronisation [1, 2, 4]. In general, the literature outlines three main schemes to deal with the multicast problem: unicast-based [1, 3], tree-based [3, 13, 17] and path-based [2, 3, 8, 14, 15]. A number of studies have shown that *path-based* algorithms exhibit superior performance characteristics over their unicast-based and tree-based counterparts, especially within *wormhole* switched networks [2, 14, 15]. In path-based multicast, when the units (called flits in wormhole switched networks) of a message reach one of the destination nodes in the multicast group, they are copied to local memory while they continue to flow through the node to reach the other destinations [2, 3, 8]. The message is removed from the network when it reaches the last destination in the multicast group.

Although many interconnection networks have been studied [3], and indeed deployed in practice, none has proved clearly superior in all roles, since the communication requirements of different applications vary widely. Nevertheless, *n*-dimensional *wormhole switched* meshes have undoubtedly been the most popular interconnection network used in practice [2, 3, 5, 6, 10, 11] due to their desirable topological properties including ease of implementation, modularity, low diameter, and ability to exploit locality exhibited by many parallel applications [3]. In wormhole switching, a message is divided into elementary units called flits, each of a few bytes for transmission and flow control. The *header* flit (containing routing information) governs the route and the remaining data flits follow it in a pipelined fashion. If a channel transmits the header of a message, it must transmit all the remaining flits of the same message before transmitting flits of another message. When the header is blocked the data flits are blocked in-situ.

Meshes are suited to a variety of applications including matrix computation, image processing and problems whose task graphs can be embedded naturally into the topology [3, 6, 10]. Wormhole switched meshes have been used in a number of real parallel machines including the Intel Paragon, MIT J-machine, Cray T3D, T3E, Caltech Mosaic, Intel Touchstone Delta, Stanford DASH [3]. Recently, among commercial multicomputers and research prototypes, Alpha 21364's multiple processors network and IBM Blue Gene uses a 3D mesh. In addition, a mesh has been recently the topology of choice for many high-performance parallel systems and local area networks such as Myrinet-based LANs. More recently, the mesh topology has been widely adopted in network-on-chip technologies, including NOSTRUM, SOCBUS, RAW (MIT) which are regular mesh architectures [11]. While our previous focused on 2D meshes, our present study investigates the multicast communication over the 3D mesh networks with the aim of generalising our previous multicast scheme [2] for higher dimensional meshes. The rest of the paper is organised as follows. Section 2 we briefly review the system. Section 3 accommodates our multicast algorithm. Section 4 conducts extensive analysis and simulation experiments and Section 5 summarises this work.

2 Preliminaries and Motivation

Definition 1: *Given a direct network composed of processors interconnected together by n-dimensional mesh topology, it can be modelled as a graph* $G = (V, E)$ *with node*

set V *and edge set* E. *In such topology, a* $N = d_{1\max} \times d_{2\max} \times d_{3\max} \times ..d_{n\max}$ *mesh:*

$V = (d_1, d_2, d_3, ..n) \in \{(d_1, d_2, d_3, ..n)_1 ..... (d_1, d_2, d_3, ..n)_N\}$:

$0 \le d_1 < d_{1\max}, 0 \le d_2 < d_{2\max}, 0 \le d_3 < d_{3\max}, ..$

$0 \le d_n < d_{n\max}, E = \{(d_{1i}, d_{2i}, d_{3i,..} d_{ni}), (d_{1j}, d_{2j}, d_{3j}, ..d_{nj})\}$, *such that any two nodes with co-ordinates* $(d_{1i,} d_{2i}, d_{3i}, ..d_{ni})$ *and* $(d_{1j,} d_{2j}, d_{3j}, ..d_{nj})$ *are connected by a communication channel if and only if*

$$\left|d_{1i} - d_{1j}\right| + \left|d_{2i} - d_{2j}\right| + \left|d_{3i} - d_{3j}\right| + ...\left|d_{ni} + d_{nj}\right| = 1$$

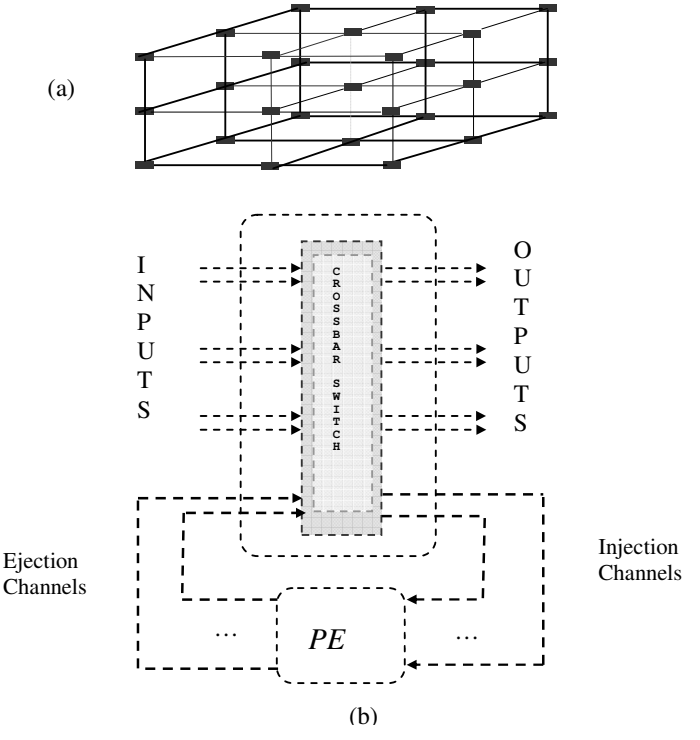

Fig. 1. (a) $3 \times 3 \times 3$ *Mesh- (b) a* node in the middle of the mesh

Fig. 1 depicts the structure of a node situated in the middle of $3 \times 3 \times 3$ mesh. The discussion can be easily extended to nodes situated at the corners and edges. A node consists of a processing element (PE) and router. The PE contains a processor and some local memory. A node uses six input and six output channels to connect to its neighbouring nodes; two in a dimension, one for each direction. There are also local channels used by the PE to inject/eject messages to/from the network, respectively.

Messages generated by the PE are injected into the network through the injection channel. Messages at the destination node are transferred to the PE through the ejection channel. The input and output channels are connected by a crossbar switch that can simultaneously connect multiple input to multiple output channels given that there is no contention over the output channels.

***Definition* 2:** *Consider a mesh* (V, E), *with node set* V *and edge set* E, *a multicast set is a couple* $(p, Đ)$, *where* $p \in V$, $Đ = \{p_1, p_2 ..., p_k\}$ *and* $p_i \in V, i = 1,...,k$. *The node* p *is the source of the multicast message, and the* k *nodes in* $Đ$ *are the destinations. To perform a multicast operation, node* p *disseminates copies of the same message to all the destinations in* $Đ$.

Existing multicast algorithms either assume a fixed network size, low dimension networks or consider only the latency at the network level. Having surveyed the literature, existing solutions to multicast communication in interconnection networks rely on two main strategies. In view of the dominance of the start-up time in the overall multicast latency, algorithms in the first strategy try to reduce the number of start-ups required to perform multicast, but this has been shown to be inefficient under high traffic loads [8, 10, 14]. For instance, the Dual Path (DP) and Multi Path (MP) algorithms proposed in [10] use this strategy. Briefly, DP uses at most two copies of the multicast message to cover the destination nodes, which are grouped into two disjoint sub-groups. This may decrease the path length for some multicast messages. The MP algorithm attempts to reduce path lengths by using up to four copies (or 2*n* for the *n*-dimensional mesh) of the multicast message. As per the multi-path multicast algorithm, all the destinations of the multicast message are grouped into four disjoint subsets such that all the destinations in a subset are in one of the four quadrants when source is viewed as the origin. Copies of the message are routed using dual-path routing (see [10] for a complete description). Algorithms in the second class, on the other hand, tend to use shorter paths, but messages can then suffer from higher latencies due to the number of start-ups required [15]. Based on this strategy, for example, the Column Path (CP) algorithm presented in [15] partitions the set of destinations into at most 2*k* subsets (e.g. *k* is the number of columns in the mesh), such that there are at most two messages directed to each column.

Generally, most existing path-based algorithms incur high multicast latency. This is due to the use of long paths required to cover the groups serially like algorithms under the umbrella of the first multicast approach or those of the second category, in which an excessive number of start-ups is involved. In addition, a common problem associated with most existing multicast algorithms is that they can overload the selected multicast path and hence cause traffic congestion. This is mainly because most existing grouping schemes [8, 10, 15] do not consider the issue of load balancing during a multicast operation. More importantly, existing multicast algorithms have been designed with a consideration paid only to the multicast latency at the network level, resulting in an erratic variation of the message arrival times at the destination nodes. As a consequence, some parallel applications cannot be performed efficiently using these algorithms, especially those applications which are sensitive to variations in the message delivery times at the nodes involved in the

multicast operation. Thus, our objective here is to propose a new multicast algorithm that can overcome the limitations of existing algorithms and thus leading to improve the performance of multicast communication in mesh networks. In our previous work [2], a new multicast scheme, the Qualified Group (QG) has been devised for meshes. Such a scheme has been studied under restricted operating conditions, such as low dimensional networks, fixed symmetric network sizes and a limited number of destination nodes [2]. In the context of the issues discussed above, this paper makes two major contributions. Firstly, the QG is generalised here with the aim of handling multicast communication in symmetric, asymmetric 3D meshes and different network sizes. Secondly, unlike many previous works, this study considers the issue of multicast latency at both the network and node levels across different traffic scenarios.

3 The Qualified Groups QG Algorithm

The QG aims at optimising the performance of message-passing communication by matching the algorithmic characteristics to the desirable properties of meshes. In other words, QG takes advantage of the partitionable structure of the mesh to divide the destination nodes into several groups of comparable sizes in order to balance the traffic load among these groups. This grouping, thus, leads to avoid the congestion problem in the network. The groups, in turn, implement multicast independently in a parallel fashion, which results in reducing the overall communication latency.

In general, the QG is composed of four phases which are described below. For the sake of the present discussion and for illustration in the diagrams, we will assume that messages are routed inside the network according to the dimension order routing. It is worth clarifying that we have adopted the dimension order routing due to the fact that this form of routing is simple and deadlock and livelock free, resulting in a faster and more compact router when the algorithm implemented in hardware, [3, 15]. However the *QG* algorithm can be used along any other underlying routing scheme, including the well-known Turn model and Duato's adaptive algorithms [3], since the grouping scheme, as explained below, in *QG* can be implemented irrespective of the underlying routing scheme (in the algorithmic level), which is not the case in most existing multicast algorithms in which destination nodes are divided based on the underlying routing used (in the routing level) [8, 10, 15]. It is worth mentioning that such a research line will be investigated further in our future works.

Phase 1: In this phase, a multicast area is defined as the smallest *n*-dimensional array that includes the source of the multicast message as well as the set of destinations. The purpose of defining this area is to confine a boundary of network resources that need to be employed during the multicast operation. The algorithm for computing the multicast area and division dimension is shown in Fig. 2.

Definition 3: *In the n-dimensional mesh with a multicast set* $(p, Đ)$ *, a multicast area* G_{MA} *includes the source node* $p[d_1, d_2, ...d_n]$ *and destination nodes* $Đ\ [(d_1, d_2, ...d_n)]$ *such that* $\forall\ d_i \in \{d_1, d_2, ..., d_n\}$*, has two corners, upper corner* $u_{d_i} = \max(Đ[d_i], p[d_i])$ *and lower corner* $l_{d_i} = \min(Đ[d_i], p[d_i])$ *such*

$$\text{that } mid_{d_i} = \begin{cases} (l_{d_i} + u_{d_i})/2 \text{ if } (l_{d_i} + u_{d_i}) \text{ is even} \\ ((l_{d_i} + u_{d_i}) - 1)/2 \text{ if } (l_{d_i} + u_{d_i}) \text{ is odd} \end{cases}$$

Procedure: the multicast area G_{MA} *and the divisor dimension* Div_{di} *in the multicast area in n-dimensional meshes.*
Input: A multicast set $(p, Đ)$ *destination nodes and a multicast message M*
Output: the multicast area G_{MA} *and the divisor dimension* Div_{d_i}

Begin

$\forall \quad d_i \in \{d_1, d_2, ..., d_n\}$

{

$u_{d_i} = \max(Đ[d_i], p_{d_i})$, $l_{d_i} = \min(Đ[d_i], p_{d_i})$ and

$$mid_{d_i} = \begin{cases} (l_{d_i} + u_{d_i})/2 \text{ if } (l_{d_i} + u_{d_i}) \text{ is even} \\ ((l_{d_i} + u_{d_i}) - 1)/2 \text{ if } (l_{d_i} + u_{d_i}) \text{ is odd} \end{cases}$$

$N_{d_i} = \left| Đ[d_i^{\Uparrow}] - Đ[d_i^{\Downarrow}] \right|$:

$Đ[d_i^{\Uparrow}] = \sum_{mid_{d_i}}^{u_{d_i}} Đ[d_i]$ and $Đ[d_i^{\Downarrow}] = \sum_{l_{d_i}}^{mid_{d_i}-1} Đ[d_i]$

}

Select the divisor dimension $Div_{d_i} = \min(N_{d_1}, N_{d_2}, ..., N_{d_n})$

Find the primary groups $= \{G_1, G_2, ... G_{g_{pr}}\}$

Fig. 2. Computing the multicast area G_{MA} and divisor dimension Div_{d_i} in *QG*

Phase 2: The multicast area G_{MA} is then divided into groups. The objective behind grouping the destination nodes is to distribute the traffic load over the multicast area in order to avoid traffic congestion, which contributes significantly to the blocking latency. Besides, grouping enables the destination nodes to receive the multicast message in comparable arrival times; i.e., this helps to keep the variance of the arrival times among the destination nodes to a minimum.

Definition 4: *In an n-dimensional mesh with a multicast set* $(p, Đ)$, *a divisor dimension* Div_{d_i} *for Đ satisfies the following condition*

$$Div_{d_i} = \min(N_{d_1}, N_{d_2}, ..., N_{d_n}), N_{d_i} = \left| Đ[d_i^{\Uparrow}] - Đ[d_i^{\Downarrow}] \right| :$$

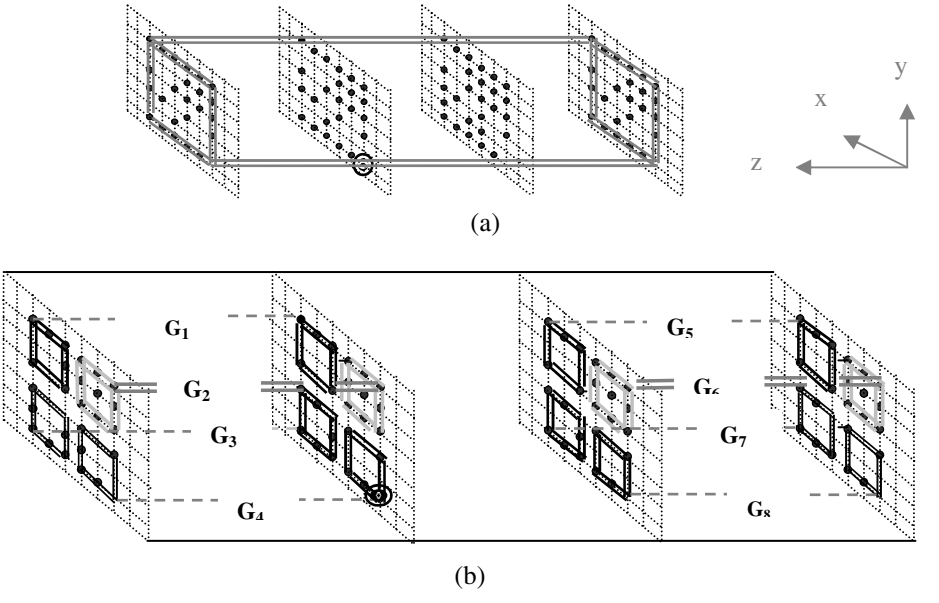

Fig. 3. (a) the multicast are in 8×10 ×4 3D mesh. (b) the grouping scheme (phase 2).

● Destination Node ◎ Source Node

$$Đ[d_i^{\Uparrow}] = \sum_{mid_{d_i}}^{u_{d_i}} Đ[d_i] \; and \; Đ[d_i^{\Downarrow}] = \sum_{l_{d_i}}^{mid_{d_i}-1} Đ[d_i]$$

Notice that if $N_{d_1} = N_{d_2}$, d_1 is given a higher priority, i.e., a higher priority is given based on the ascending order of the dimensions. For instance, if $N_x = N_y = N_z$, X dimension will be considered as a divisor dimension. The divisor dimension is used as a major axis for the grouping scheme in this phase. The multicast area G_{MA} is then divided into a number of disjoint groups as formulated in the following definition.

Definition 5: *Given an n-dimensional mesh with a multicast set* $(p, Đ)$ *and a multicast area* G_{MA}, $\forall G_i, G_j : G_i \subseteq G_{MA}$ *and* $G_j \subseteq G_{MA} \rightarrow G_i \cap G_j = \Phi$.

According to Definition 5, G_{MA} is divided into a number of primary groups as given in equation 1; where g_{pr} refers to the number of primary groups obtained after dividing the destination nodes over the division dimension, such that

$$g_{pr} = \begin{cases} p_t & \text{if } \exists G_i \subseteq G_{MA} : G_i = \Phi \\ 2^n & \text{otherwise} \end{cases} \quad (1)$$

where p_t is an integer, $1 \le p_t < 2^n$

For the sake of illustration, let the system be a 8×10 ×4 3D mesh, the multicast area is determined as depicted in Fig. 3.(a), the division dimension is Z and the destinations have been divided into 8 groups as illustrated in Fig. 3(b).

Phase 3: This phase is responsible for qualifying the groups already obtained in the preceding phase for a final grouping. Having obtained the primary groups, g_{pr}, we recursively find the multicast area for each group, $G_i \subseteq G_{MA}$, as defined in Definition 4, and determine the internal distance $Int(G_i)$ for each group G_i.

$$Int(G_i) = Dist(p_f(G_i), p_n(G_i)) + N_{G_i} \tag{2}$$

Where $Dist$ refers to the Manhattan distance in which the distance between tow nodes, for instance the distance between two nodes $(p1_x, p1_y)$ and $(p2_x, p2_y)$ is given by $Dist(p1, p2) = |(p1_x - p2_x)| + |(p1_y - p2_y)|$. While the first term, $Dist(p_f(G_i), p_n(G_i))$, in the above equation represents the distance between the farthest p_f and the nearest node p_n in a group G_i from/to the source node p, respectively, the second term, N_{G_i}, represents the number of destination nodes that belong to the relevant group $G_i \subseteq G_{MA}$. We then determine the external distance $Ext(G_i)$.

$$Ext(G_i) = Dist(p_n(G_i), p) \tag{3}$$

The minimum weight W_m for a group $G_i, 1 < i \le g_{pr}$, where g_{pr} refers to the number of primary groups, is then calculated by

$$W_m(G_i) = Ext(G_i) + Int(G_i) \tag{4}$$

Definition 6: *Given a multicast area G_{MA} and $G_i \subseteq G_{MA}$, where $1 < i \le g_{pr}$, the average of the minimum weights W_{av}, for the multicast area G_{MA}, is given by*

$$W_{av} = \frac{\sum_{i=1}^{g_{pr}} W_m(G_i)}{g_{pr}} \tag{5}$$

Definition 7: *Given a multicast area G_{MA}, $G_i \subseteq G_{MA}$, and W_{av}, the qualification point, $QP(G_i)$, for each group is calculated as follows*

$$QP(G_i) = \frac{(W_m(G_i) - W_{av})}{W_{av}} \tag{6}$$

The qualification point for each group is compared to an assumed threshold value TD, which is used to set a limit for the partitioning process.

Definition 8: *Given a multicast area* G_{MA} *and* $G_i \subseteq G_{MA}$, *we say that* G_i *is a qualified group if and only if its minimum weight* $W_m(G_i) \leq W_{av}$ *or if its qualification point* $(QP(G_i)) \leq TD$.

For example, given that the threshold value is $TD = 0.5$, each qualified group must hold at least half of the total average weight W_{av} of the groups. Once a group $G_i \subseteq G_{MA}$ does not satisfy the condition formulated in Definition 8, it is treated as an unqualified group. In this case, this unqualified group is divided into two sub-groups based on its division dimension.

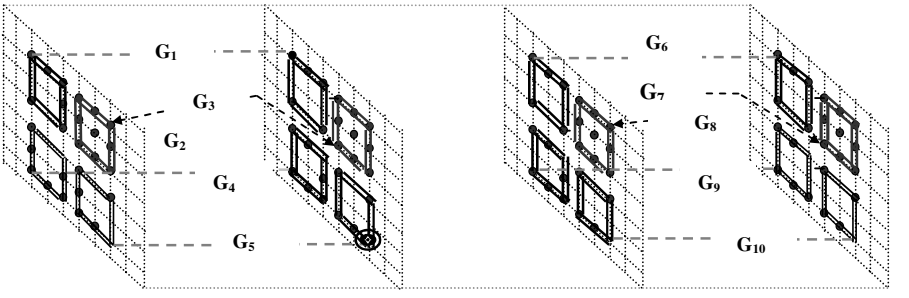

Fig. 4. The qualified groups in 8×10 ×4 3D mesh (Phase 3)

Following the example shown in Fig.3 and using a threshold value $TD = 0.5$, we find in this example that both G_2 and G_6 (as shown in Fig. 3) are not qualified (based on Definition 8). Therefore, the multicast area for each group (G_2 and G_6) is divided into two further sub-groups based on the division dimension (Z in this case) as depicted in Fig. 4. The new sub-groups are then compared to the qualified groups already obtained. After qualifying all the groups, the source node sends the message to the representative nodes in the qualified groups. If the new resulting groups are qualified the partitioning process is terminated. Otherwise, the unqualified group is divided into a number of sub-groups sb, where $2 \leq sb \leq 2^n$. For instance, for any unqualified group $G_i \subseteq G_{MA}$ in the 3D mesh, it can be divided into 8 groups at maximum, even if the new obtained groups are still larger than those which meet the qualification point. In fact, the partitioning process is terminated at this stage in order to reduce the number of comparisons during the qualifying phase. This helps to keep the algorithm simple and maintains a low preparation time.

Phase 4: For each group resulting from Phase 3, the nodes which have the lowest communication cost, in terms of distance are selected as the representative nodes of the qualified groups that can receive the multicast message from the source node. In other words, the nearest node for each qualified group is elected so that it could be sent the multicast message with a single start-up only. Concurrently, the representative nodes act as "source" nodes by delivering the message to the rest of the destination nodes in their own groups with one additional start-up time only.

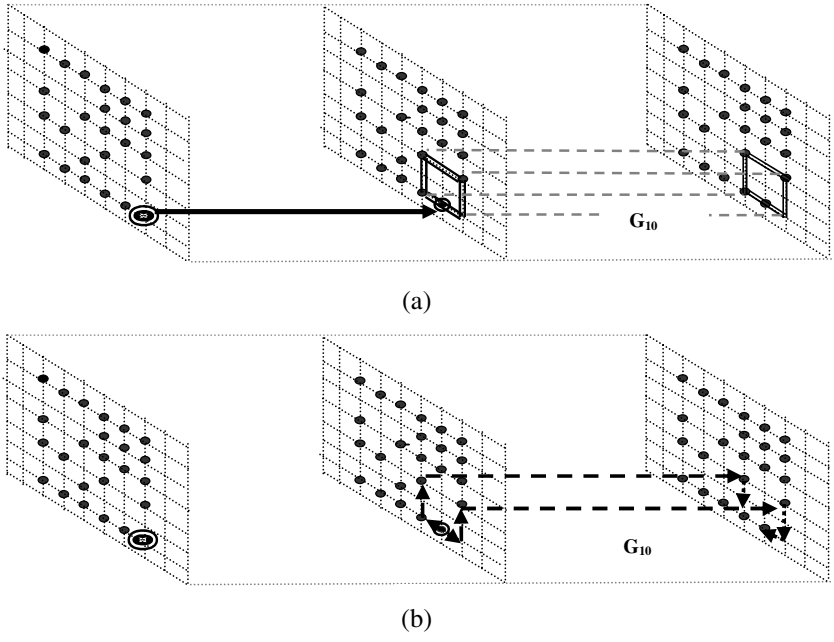

Fig. 5. (a) The first communications step (which occurs in phase 4) in the QG multicast algorithm, (b), The second communications step (which occurs in phase 4)

After qualifying all the groups, the source node sends the message to the representative nodes in the qualified groups. The source node performs this operation with a single start-up latency taking advantage of the multiple-port facility of the system by creating two disjoint paths in this step. For the sake of clarity, we have selected group G_{10} to represent this step as illustrated in Fig. 5(a), where the source node sends the message to the selected representative node. Concurrently, every representative node in each group acts as a source node and, in turn, sends the message to the rest of the destinations in its own group as the representative node does in Fig. 5(b).

4 Performance Evaluation

A number of simulation experiments have been conducted to analyse the performance of *QG* against *DP*, *MP* and *CP*. A simulation program has been developed to model the multicast operation in the mesh. The developed model has been added to a larger simulator called MultiSim [6], which has been designed to study the collective communication operations on multicomputers and has been widely used in the literature [2, 10, 12]. The simulation program was written in VC++ and built on top the event-driven CSIM-package [7]. We have used the 2D mesh with four injection channels and four ejection channels. Two unidirectional channels exist between each pair of neighbouring nodes. Each channel has a single queue of messages waiting for

Table 1. The coefficient of variation of the multicast latency in the DP, MP and CP algorithms with the improvement obtained by QG (QG_{IMPR} %) in the 10×10×10 3D mesh

	#Destinations=20		# Destinations=60		# Destinations=80	
	CV	(QG_{IMPR} %)	CV	(QG_{IMPR} %)	CV	(QG_{IMPR} %)
DP	0.4156	40.01	0.5056	62.70	0.5678	78.62
MP	0.3605	21.96	0.4710	51.77	0.4873	53.14
CP	0.4967	67.34	0.5590	79.74	0.5925	86.48
QG	CV= 0.2967		CV= 0.3107		CV = 0.3176	

transmission. In our simulations, the start-up latency has been set at 33 cycles, the channel transmission time at 1 cycle and the threshold *TD* at 0.5.

The network cycle time in the simulator is defined as the transmission time of a single flit across a channel The preparation time (which consists of dividing the destination nodes into appropriate subsets and creating multiple copies of the message as needed, depending on the underlying algorithm) of the DP, MP, CP and QG algorithms are set at 2, 2, 4 and 16 cycles, respectively. The preparation time was deliberately set higher in the QG algorithm to reflect the fact that our algorithm requires a longer time to divide the destinations into qualified groups. All simulations were executed using 95% confidence intervals (when confidence interval was smaller than 5% of the mean). The technique used to calculate confidence intervals is called batch means analysis. In batch means method, a long run is divided into a set of fixed size batches, computing a separate sample mean for each batch, and using these batches to compute the grand mean and the confidence interval. In our simulations, the grand means are obtained along with several values, including confidence interval and relative errors which are not shown in the figures. Like existing studies [1, 2, 3, 10, 15, 13], only the grand mean is shown in our figures.

4.1 Latency at the Node Level and Average Additional Traffic

This section presents the coefficient of variation of the multicast latency as a new performance metric in order to reflect the degree of parallelism achieved by the multicast algorithms. A set of simulation experiments have been conducted where the message inter-arrival times between two messages generated at a source node is set at 250 cycles. The message length is fixed at 64 flits and the number of destination nodes is varied from 20, 30, 40… to 60 nodes. The coefficient of variation (CV) is defined as SD / M_{nl}, where SD refers to the standard deviation of the multicast latency (which is also the message arrival times among the destination nodes) and M_{nl} is the mean multicast latency. The coefficient of variation of *QG* has been compared against that of *DP, MP* and *CP*. Table 1 contains performance results for the $10 \times 10 \times 10$ mesh, which have been obtained by averaging values obtained from

at least 40 experiments in each case. The $QG_{IMPR}\%$ in Table 1 refers to the percentage improvement obtained by *QG* over its *DP, MP* and *CP* competitors.
As shown in Table 1, *QG* achieves a significant improvement over *DP, MP* and *CP*. This is due firstly to the efficient grouping scheme adopted by *QG* which divides the destinations into groups of comparable sizes. Secondly, and more importantly, unlike in *DP, MP* and *CP*, the destination nodes for each qualified group in *QG* (except those selected in the first message-passing step) receive the multicast message in the second message-passing step, in parallel. This has the net effect of minimising the variance of the arrival times at the node level. In contrast, *DP, MP* and *CP* perform multicast with either longer paths as in *DP* and *MP* or in an excessive number of message-passing steps, as in *CP*.

The additional traffic is computed as in [8, 10], that is, by subtracting the number of destination nodes from the number of channels involved in the multicast operation. This reflects the amount of network resources that are used to complete a multicast

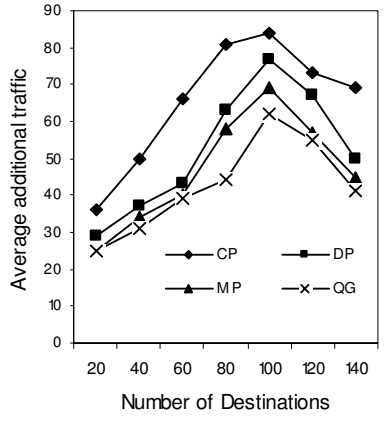

Fig. 6. Average additional traffic as a function of the number of destinations for the four algorithms, DP, MP, CP and QG in 8×8 ×8 mesh

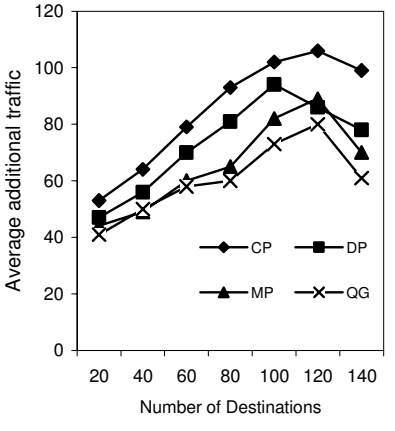

Fig. 7. Average additional traffic as a function of the number of destinations for the four algorithms, DP, MP, CP and QG in 10×10 ×10 mesh

operation. A physical channel occupied for one cycle is considered as one-traffic unit. Figs. 6 and 7 shows the resulting average additional traffic in the four algorithms for various numbers of destination nodes and two different network sizes, 8×8 ×8 and 10×10 ×10 3D meshes, respectively. To complete a multicast operation, *QG* requires fewer channels than *DP*, *MP* and *CP* since the destinations are divided into several groups which are reached in a more efficient manner.

4.2 Latency in the Presence of Multicast and Unicast Traffic

In some real parallel applications, a message may have to compete for network resources with other multicast messages or even with other unicast messages. To

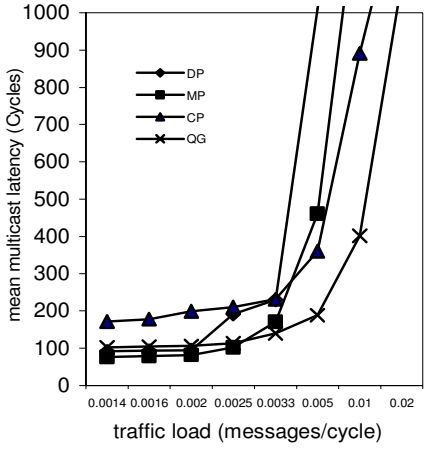

Fig. 8. Mean multicast latency in the 8×8×8 mesh. Message length is 64 flits, number of destination =10, traffic consists of multicast (10%) and unicast (90%).

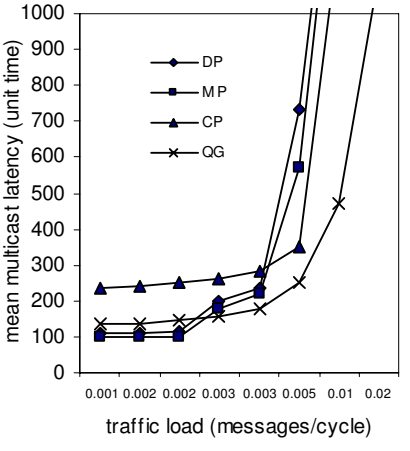

Fig. 9. Mean multicast latency in the 8×8×8 mesh. Message length is 64 flits, number of destination 20 nodes, traffic consists of multicast (10%) and unicast (90%)

examine performance in such situation, results for the mean multicast latency have been gathered in the $8\times8\times8$ mesh in the presence of both multicast (10%) and unicast (90%) traffic (similar studies are outlined in [8, 10, 15]). The message size is set at 64 flits and the number of destinations in a given multicast operation has been set to 10 and 20 nodes, respectively. The simulation results are provided in Figs. 8 and 9. Fig. 8 reports results for 10 destinations while Fig. 9 shows results for 20 destinations. Under light traffic, QG, DP and MP have comparable performance behaviour, with MP having a slightly lower latency. On the other hand, CP has a higher time. This is mainly due to the dominating effect of the start-up latency in such a situation. However, under heavy traffic, an opposite behaviour is noticed in that QG performs the best in terms of both latency and throughput, followed by CP. More importantly, we can observe from Fig. 9 that as the number of destinations increases the performance advantage of QG becomes more noticeable over that of CP. This is mainly because QG alleviates significantly the congestion problem at the source node. In contrast, the source node in CP suffers from a higher load and as more destinations are involved in the multicast operation, the more severe this limitation becomes.

5 Conclusions and Future Directions

In this study, the QG multicast algorithm has been generalised for n-dimensional meshes. In this paper, 3D meshes have been considered in our performance evaluation. Results from simulations under different conditions have revealed that the *QG* algorithm exhibits superior performance over well-known algorithms, such as dual-path, multiple-path, and column-path algorithms. Unlike existing multicast

algorithms, the *QG* algorithm can maintain a lower variance of message arrival times at the node level. Consequently, most of the destination nodes receive the multicast message in comparable arrival times. Our Results show also that the QG has improved the scalability of the multicast operation in 3D meshes. It would be interesting to further investigate the interaction between the important parameters that affect the performance of the *QG* algorithm, notably the grouping scheme, network size, threshold value, multicast group size, and traffic load, with the aim of proposing an analytical model that could predict, for example, the multicast latency given a particular grouping scheme, network size, multicast group size, and traffic load. Moreover, another possible research line is to apply this multicast scheme on Network-on-Chip platforms.

References

1. Wang, N.-C., Yen, C.-P., Chu, C.-P.: Multicast communication in wormhole-routed symmetric networks with hamiltonian cycle model. Journal of Systems Architecture 51(3), 165–183 (2005)
2. Al-Dubai, A., Ould-Khaoua, M., Mackenzie, L.: An efficient path-based multicast algorithm for mesh networks. In: Proc. the 17th Int. Parallel and Distributed Processing Symposium (IEEE/ACM-IPDPS), Nice, France, April 22 -26, pp. 283–290 (2003)
3. Duato, J., Yalamanchili, C., Ni, L.: Interconnection networks: an engineering approach. Elsevier Science, Amsterdam (2003)
4. Touzene, A.: Optimal all-ports collective communication algorithms for the k-ary n-cube interconnection networks. Journal of Systems Architecture 50(4), 169–236 (2004)
5. Chen, Y.-S., Chiang, C.-Y., Chen, C.-Y.: Multi-node broadcasting in all-ported 3-D wormhole-routed torus using an aggregation-then-distribution strategy. Journal of Systems Architecture 50(9), 575–589 (2004)
6. McKinley, P.K., Trefftz, C.: MultiSim: A simulation tool for the study of large-scale multiprocessors. In: MASCOTS' 1993. Proceedings of the Int. Symp. Modeling, Analysis and Simulation of Computer and Telecommunication Systems, pp. 57–62 (1993)
7. CSIM: A C-based, process-oriented simulation language http://www.mesquite.com/
8. Fleury, E., Fraigniaud, P.: Strategies for path-based multicasting in wormhole-routed meshes. J. Parallel & Distributed Computing 60, 26–62 (1998)
9. Tseng, Y.-C., Wang, S.-Y., Ho, C.-W.: Efficient broadcasting in wormhole-routed multicomputers: A network-partitioning approach. IEEE Transactions on Parallel and Distributed Systems 10(1), 44–61 (1999)
10. Lin, X., McKinley, P., Ni, L.M.: Deadlock-free multicast wormhole routing in 2D-mesh multicomputers. IEEE Transactions on Parallel and Distributed Systems 5(8), 793–804 (1994)
11. Bjerregaard, T., Mahadevan, S.: A survey of research and practices of Network-on-Chip. ACM Computing Surveys 38, 1–51 (2006)
12. Robinson, D.F., McKinley, P.K., Cheng, C.: Path based multicast communication in wormhole routed unidirectional torus networks. Journal of Parallel Distributed Computing 45, 104–121 (1997)
13. Malumbres, M.P., Duato, J.: An efficient implementation of tree-based multicast routing for distributed shared-memory multiprocessors. J. Systems Architecture 46, 1019–1032 (2000)

14. Mohapatra, P., Varavithya, V.: A hardware multicast routing algorithm for two dimensional meshes. In: The Eighth IEEE Symposium on Parallel and Distributed Processing, News Orleans, pp. 198–205 (1996)
15. Boppana, R.V., Chalasani, S., Raghavendra, C.S: Resource deadlock and performance of wormhole multicast routing algorithms. IEEE Transactions on Parallel and Distributed Systems 9(6), 535–549 (1998)
16. Wang, S., Tseng, Y., Shiu, C., Sheu, J.: Balancing traffic load for multi-node multicast in a wormhole 2D torus/mesh. The Computer Journal 44(5), 354–367 (2001)
17. Kumar, D.R., Najjar, W.A., Srimani, P.K.: A new adaptive hardware tree-based multicast routing in k-ary n-cubes. IEEE Computer 50(7), 647–659 (2001)