



DBM: A Dimension-Bubble-Based Multicast Routing Algorithm for 2D Mesh Network-on-Chips

Canwen Xiao, Hui Lou, Cunlu Li^(✉), and Kang Jin

College of Computer, National University of Defense Technology,
Changsha 410073, Hunan, China
{cwxiao, huilou, cunluli, kangjin}@nudt.edu.cn

Abstract. Network-on-Chips (NoCs) has been widely used today for efficient communication in multicore systems. Existing NoCs mostly use 2D mesh topology in commercial and experimental manycore processors since it maps well to the 2D layout. For 2D mesh, dimension order routing and different adaptive routing algorithms performs well in unicast traffic but suffer from poor performance when faced with one-to-many (multicast) traffic. Efficient multicast routing algorithm is an important target for the design of special on-chip networks such as neural networks. Recently proposed multicast routing algorithms are less efficient or can introduce unbalanced load in some situations. In this paper, we propose DBM, a novel multicast routing algorithm based on the dimension-bubble flow control for 2D mesh networks. DBM is deadlock-free while achieving the minimal path and fully-adaptive multicast routing algorithm. Moreover, DBM simplifies the deadlock condition where the escape channel is not necessary. Evaluation results show that DBM can achieve much better performance than existing multicast routing algorithms, with 18% reduction in packet latency and 16% improvement in network throughput.

Keywords: Dimension-bubble flow control · Multicast routing algorithm · Deadlock

1 Introduction

Network-on-Chips (NoCs) has always been a challenging research topic, providing a scalable solution for Multiprocessor System-on-Chip (MPSoC). 2D mesh topology is usually preferred due to its layout on a planar surface in the chip. The topology of a 4-ary 2-cube mesh and corresponding router microarchitecture are presented in Fig. 1.

In addition to unicast communication, NoCs also needs to deal with a lot of multicast communication [3]. Multicast messages are useful for efficient execution of parallel programs as the multicast communication is frequently employed in many MPSoC applications such as replication [8], barrier synchronization [13],

cache coherency in distributed shared-memory architectures [6] and clock synchronization [1]. In these MPSoC applications, it is a key issue to ensure efficient communication for multicast packets. On the other hand, the number of processor cores integrated on the chip is also increasing. For example, SpiNNaker project aims to produce 10,000-core chips for modeling of large-scale spiking neural networks in biological real time [9]. For these million processor machines, multicast packets with appropriate multicast routing algorithms can effectively reduce the number of packets in the network to alleviate network congestion.

Some theories and methodologies have been proposed [4, 7, 10, 11] to achieve deadlock-free multicast routing. Virtual circuit tree multicasting (VCTM) [4], as a representative, achieves a tree-based routing algorithm to support multicasting in NoCs. VCTM builds several virtual circuit trees through the destinations before the multicast messages are injected into the network. VCTM achieves this scheme by sending separate unicast setup messages (look ahead signals) for each destination, through the utilization of virtual circuit table (VCT) and content addressable memory.

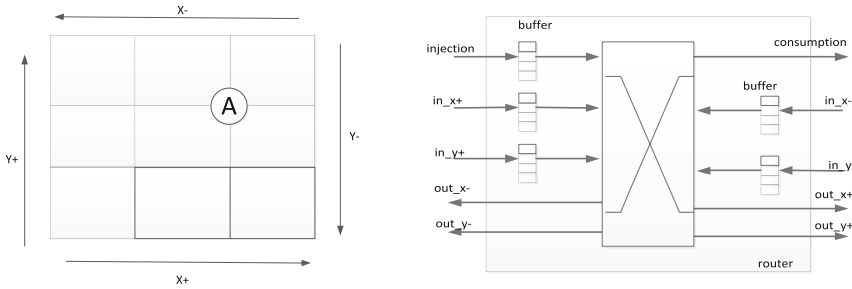


Fig. 1. 4-ary 2-cube mesh

In VCTM, cyclic dependencies can be avoided by using the Dimension Order Routing (DOR) algorithm for both the setup and the multicast messages. However, some shortcomings can be introduced within VCTM. First, VCTM's design complexity and hardware overhead strongly depends on the network size, making it difficult to scale up. Second, VCTM is less efficient when faced with high injection rate network conditions. Third, when updating the VCT, the source node has to send discrete unicast setup messages per destination. In this situation, when faced with large number of destinations, the number of unicast setup message will be increased, thereby reducing the performance. Recursive Partitioning Multicast (RPM) [11] is another representative multicast routing algorithm. In RPM method, the processing of the header information is complex and will be performed several times for each multicast message. VCTM and RPM share the same disadvantage that a message may hold several output channels, thereby increasing network contention. Finally, both RPM and VCTM are based on deterministic algorithms and cannot provide adaptiveness to neither unicast

nor multicast messages. Paper [7] presents a routing algorithm called Balanced Adaptive Multicast (BAM). This algorithm adopted Duato's principle [2] to realize the deadlock-free adaptive routing.

In our former works, the Dimensional Bubble Routing Algorithm (DBRA) [12] is proposed for Mesh networks. This algorithm realizes a fully adaptive routing for unicast communication without the escape channels. In this paper, DBM, a novel dimension-bubble-based multicast routing algorithm is studied based on the idea of DBRA. The contributions of this paper are as follows:

1. The strategy of dimensional-bubble flow control is presented for multicasting operation in 2D Mesh networks and the novel multicast routing algorithm, DBM, is studied;
2. We proof and present that DBM is deadlock-free with efficient multicast communication;
3. We provide a thorough evaluation of the proposed organization and demonstrate that we can achieve higher performance.

The rest of this paper is organized as follows. In Sect. 2, the novel multicast routing algorithm will be presented. In Sect. 3, we prove that the proposed flow control strategy can ensure that the minimal path and fully-adaptive routing algorithm is deadlock-free. The performance of the novel algorithm is evaluated in the Sect. 4. In the end, we summary this paper in Sect. 5.

2 Novel Multicast Routing Algorithm

In this section, we design DBM, a novel multicast routing algorithm based on the dimensional-bubble flow control. Firstly, the algorithm schemes of RPM and BAM will be analyzed and the novel multicast routing algorithm will be presented based on the study of RPM and BAM.

2.1 RPM and BAM Routing Algorithm

RPM algorithm uses the determinate method to divide the network to eight regions according to the router's location. Then, according to the destinations of the multicast packet, the output port of packet will be calculated by deterministic rule. RPM algorithm ensures the multicast packets are transmitted along the same path as more as possible. At the same time, RPM also strives to balance the load of network. Figure 2 depicted an example of eight regions of RPM in 4-ary 2-cube Mesh network.

RPM algorithm adopts two virtual networks called VN0 and VN1 to avoid the existence of deadlock routing in the network. However, this method can bring unbalanced network communication to degrade the performance. Similarly, BAM algorithm also divides the network to eight regions depended on the location of the multicast packet. BAM multicast routing algorithm is based on the strategy of full-adaptive routing of Duato's principle, and choose the output port with lower buffer utilization when there exist two or more available output ports.

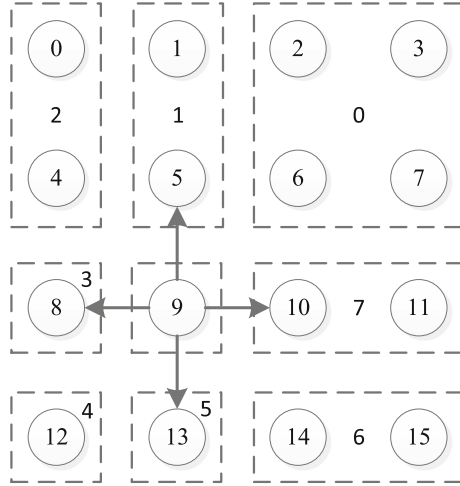


Fig. 2. An example of eight regions of RPM

2.2 Dimension-Bubble Multicast (DBM) Algorithm

We propose the novel multicast algorithm called Dimension-Bubble Multicast (DBM) based on the study of RPM and BAM algorithm. At the same time, DBM algorithm adopts the strategy of minimal path and realizes the multicast routing based on the idea of DBRA algorithm [12].

In DBRA, the definition of dimensionbubble flow control for unicast communication is as follows:

when a packet wants to move to the next buffer, if there are remaining routing hops in N dimensional directions ($N \leq n$), then this packet can request for arbitration of the next buffer only when there are more than or equal to N free packet spaces in the next buffer. Otherwise, it has to wait.

DBRA routing algorithm uses the remaining number of hops in a dimension to judge the next step of packet’s routing. In order to support the multicast routing, we propose a new strategy of flow control based on DBRA in 2-dimensional (2D) Mesh networks.

In 2D Mesh networks, we define the set of destination nodes of a multicast packet as

$$\{D_1, D_2, \dots, D_i, \dots, D_n | 1 \leq i \leq n, n \text{ is an integer}\}$$

Suppose that the packet needs to transmit M_i dimensional directions before arriving at the destination node D_i . We define $\text{Max}\{M_i\}$ to represent the maximum value in the set of $\{M_1, M_2, \dots, M_i, \dots, M_n | 1 \leq i \leq n, n \text{ is an integer}\}$.

Based on the above definitions, the novel flow control strategy can be described as follows:

The multicast packet can request for arbitration of the next buffer only when there exist more than or equal to $\text{Max}\{M_i\}$ free packet space in the input buffer of the next-hop router. Otherwise, it has to wait.

We name the new flow control strategy as DBMFC (Dimensional Bubble Multicast flow control). For 2D Mesh networks, once a multicast packet arrived at the input buffer of a router, it may have remaining routing paths in $X+/X-$, $Y+/Y-$ dimensional directions. But if only generic minimal path can be chosen in 2D Mesh networks, each destination node has routing hops only in two dimensions at most.

Therefore, $\text{Max}\{M_i\} \leq 2$ can be guaranteed in 2D Mesh networks, and thus it is enough for the input buffer to be set with 2-packet size, which can meet the demand of DBMFC for multicast operations.

Based on DBMFC, we propose DBM, to achieve a fully-adaptive multicast routing algorithm. The following is the description of DBM algorithm:

- Firstly, minimal-path routing is adopted as the baseline in 2D mesh networks. For each multicast packet in the input buffer, it is calculated how many hops this multicast packet must transmit on the different dimensions for each destination node.
- Secondly, multicast packets calculate the remaining dimensions of the destination nodes and appeal the arbitration requests of the different buffers meet with the strategy of DBMFC.
- Thirdly, if the number of granted requests is more than one, DBM algorithm will choose an output port with lower buffer utilization in the next-hop router. This process is similar to RPM and BAM algorithms.
- Fourthly, the replicated packet carried the information of those destination nodes had remaining hops in the dimensional direction to flow out. At the same time, the number of hop of destination node in this dimensional direction minus 1.
- The multicast packets repeat from step 1 to step 4 until all the destination nodes have been traversed.

Compared with DBRA algorithm for unicast routing, DBM algorithm decides the next routing dimensional direction by the value of $\text{Max}\{M_i\}$. The choosing strategy of arbitration of DBM algorithm is similar as the arbitration strategy of RPM and BAM based on regions and the entire network is divided into eight regions labelled as 0, 1, 2, 3, 4, 5, 6 and 7 such as Fig. 2.

We explain the routing strategy of DBM algorithm in Fig. 2. Suppose that the multicast packet hops on the $Y+$ direction. When all destination nodes of packet locate in the region $Y+$ (region 1 in Fig. 2), it means that the set of destination nodes do not include those nodes in region 0 and 2.

If there are free spaces in the next router of $Y+$ direction, the packet may enter to the buffer of the next router. On the other hand, if there are nodes of region 0 or region 2 in the destinations of packet, the packet may enter when there are more than or equal to 2 free packet spaces of the next router in $Y+$ direction.

For DBM routing algorithm, asynchronous replication is adopted. In asynchronous replication, branch replicas will not block each other, since each of them proceeds independently. It means that replicated packets can be granted in different directions respectively.

3 Proof of Deadlock Freedom of DBM

The goal of this section is to explain how the DBM algorithm achieves the dead-lock free characteristic for any minimal path, adaptive routing on 2D mesh networks.

DBM algorithm is achieved based on DBRA algorithm which is fully adaptive routing for unicast operations. Since the deadlock freedom of DBRA has been proved in paper [12], if the differences between DBM and DBRA do not cause dead lock of network, it can be concluded that DBM algorithm is deadlock freedom.

The differences between DBRA and DBM are that multicast packets may carry with more than one addresses of destination nodes and the replication operation of packet will be performed in the router and it will increase the number of packets in the network. We should analyze whether the replication operation will cause dead lock in the network.

Proof Sketch: In each step of the proof, we will analyze all possible cases of packet in the network and present allocation of buffers to prove that all kinds of packets can reach the destination node. Accordingly, the conclusion that the deadlock does not exist in the network can be made as a result.

Proof. If the replication operation was committed before multicast packets injecting into the network, this replication operation cannot cause the dead lock in the network. This is because the replicated packets will be injected into the network as same as the unicast packets,

Next, we analyze the replication operation after multicast packets have been injected into the network. In this situation, the multicast packets may be stored in the buffer of one-dimensional direction such as $X+$, $X-$, $Y+$ and $Y-$ in the 2D mesh networks. Without loss of generality, we assume multicast packet is in the buffer of direction $X+$. The multicast packets on buffer space have three cases:

- The first case is that remaining hops of all destination nodes of multicast packet are in the direction $X+$, it means that the value of $\text{Max}\{M_i\}$ of this packet is 1.
Suppose: the space of buffer can contain two packets. The situation of packet is described by the third graph.

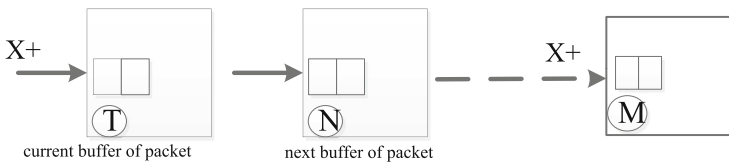


Fig. 3. Buffers in $X+$

There exist two possible cases for the next buffer. If there is one free space in the next buffer, according to the flow control strategy of DBMFC, since the value of $\text{Max}\{M_i\}$ is equal to 1, the replicated packets can enter the next buffer. If there is not room in the next buffer, we can conclude that there is no less than one packet in the next buffer whose destination nodes are only in $X+$ or packet is waiting for consumption in the next buffer. Suppose that the destination nodes of two packets in the next buffer remain hops in two directions or one direction that is not $X+$, According to the flow control strategy of DBMFC, when $\text{Max}\{M_i\}$ is equal to 2, it is necessary that the destination buffer must own two free packets space at least. So, it is impossible that the destination nodes of two packets in the next buffer remain hops in two directions or one direction that is not $X+$. We can reach the conclusion that there is no less than one packet in the next buffer whose destination nodes are only in $X+$ or packet s waiting for consumption in the next buffer. The situation of the latter buffer is the same as the next buffer.

Because there are not wraparound connections in X and Y direction, the cyclic dependency cannot be formed in X or Y direction. As a consequence, the forever block is not formed between packets of this case. Because it is impossible that the destination nodes of packets in the last buffer in $X+$ direction need to hop in $X+$ direction, it is true that packet waiting for consumption exist in the last buffer. The packet o will be consumed soon, thus, these replicated packets of the first case can always move and reach the destination node.

- The second case is that it is existed that the destination nods of multicast packet in buffer have only one-direction routing and different direction with the buffer space where the packet is placed. We analyze the possibility of packet of destination's buffer when there is not room in the destination's buffer. Suppose that the packets that are occupying the destination's buffer space are the packets waiting for consumption or the packets of the first case. Because the packets of two cases can always go ahead from the current buffer, they will not block the packets of the second case forever. Suppose that the packets that are occupying the destination's buffer space are those packets whose destination nodes remain X and Y direction routing or are also packets of the second case. According to DBMFC, these packets cannot occupy all buffer space and the destination buffering must remain one free packet space after they enter the destination buffer. Thus, the replicated packets of the second case can enter the destination's buffer space. So, the packets of the second case can always reach the destination nodes.
- The last case is that it is existed that the destination nods of multicast packet in buffer remains X and Y direction routing. According to DBM algorithm, the replicated packet remaining X and Y direction routing can have requests in X and Y direction at the same time. We consider the situation of those buffers which have the same direction with the buffer in which the packet is placed at present. Suppose the direction of current buffer space is $X+$. The situation of buffer is the same as the Fig. 3.

According to the above analyses, if the packets that are occupying the next buffer space are those packets of the above two cases or waiting for consumption, they cannot result in other packets blocking forever. As a result, the deadlock maybe only exists among the packets of the last case. According to DBM algorithm, it is certain that those packets of the last buffer in $X+$ direction have one-direction routing at most, because they have the routing in $X+$ direction no longer. These packets are the packets of the former two cases or waiting for consumption. Because the packets in the buffer space in $X+$ direction can move certainly, the packets of the last case may finish the routing of one-direction and become the packets of the former two cases or waiting for consumption. So, the packets of the last case will also reach the destination node.

Based on the above proof, we can conclude that DBM will not introduce deadlock that when the scheme is deadlock free in the 2D mesh.

4 Evaluation

In this section, we study performance and scalability of DBM algorithm supporting fully-adaptive multicast routing. The DBM algorithm is common minimal path, fully-adaptive routing algorithm except for DBMFC. We base our evaluation on the BookSim simulator [5] developed at the Stanford University, thanks to its modular design, and the availability of a large variety of classic network implementations. DBM algorithm was implemented in the BookSim simulator with little effort.

We compare average packet latencies of DBM to its counterparts: RPM and BAM algorithms. More specifically, for BAM algorithm, Duato’s method reserves one virtual channel, which employs dimension order routing (DOR), as an escape channel, and the other virtual channels employ the shortest path adaptive routing algorithm.

4.1 Experiment Setup

It is noteworthy to mention that in our design, DBM takes the same way as DBRA to allow a packet to be chosen for the arbitration independently, without considering its position in the buffer. In other words, head of line (HOL) blocking will not occur in DBM. To ensure fair comparisons, we have removed HOL blocking from the implementations of RPM and BAM, which also stresses the effect of multicast routing strategies. In addition, we have assigned the same amount of buffer space to each router in the evaluation, and in the case of BAM, uses exactly the same adaptive routing function as DBM.

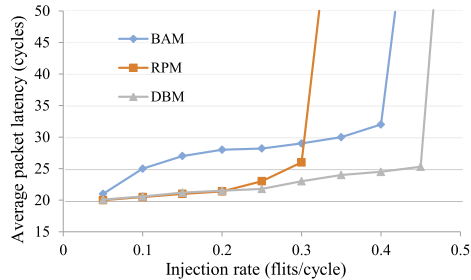
The simulator is warmed up for 10,000 cycles and then the performance is measured over another 100,000 cycles. Network is considered unstable when average latency of one packet exceeds 1000 cycles [5], therefore we stop reporting results when latency is beyond this point. Flit injection rate used in the simulation is defined as the time to take a single flit to be injected at a source. For example, injection rate is 0.1 means that each source injects a new flit in one out of every ten simulator cycles. Our simulation settings are summarized in Table 1.

Table 1. BookSim configuration parameters

Configuration parameters	Value
Topology	4-ary 2-cube mesh, 8-ary 2-cube mesh
Number of VCs	4, 6, 8
Proportion of multicast packets	10%, 15%, 20%
Injection rate (flits/cycle/node)	0.1–0.5
Number of destinations in multicast packets	6, 9, 12, 15
Traffic pattern	Uniform random, random permutation, bit rotation, transpose

4.2 Average Latency and Load Scalability

The Fig. 4, 5, 6, 7 plots the average latencies for 2-cube mesh network under different patterns. DBM outperforms the other two algorithms in all cases. In particular, under the pattern of uniform random, the saturation throughput of DBM algorithm is largest in three algorithms and BAM is better than RPM algorithm. When the load of network is low, the latency of three algorithms is almost same. However, with the increase of load of the network, the performance of RPM is the worst. It shows that the load of two virtual networks in RPM algorithm is unbalance. Compared with BAM, DBM algorithm achieved 8.6% latency reduction and 6.3% throughput improvement.

**Fig. 4.** Average packet latency in uniform random traffic

Similarly, in transpose and random permutation patterns, the saturation throughput of DBM is the largest and BAM is the following. The advantage of performance of DBM is obvious. Relative to RPM, DBM achieved 24.9% and 12.7% latency reduction and 71.4% and 75% throughput improvement respectively. Since the traffic patterns of transpose and random permutation cause the unbalancing load of different dimensions easier than uniform random, for RPM algorithm, the performance in these patterns is worse than uniform random.

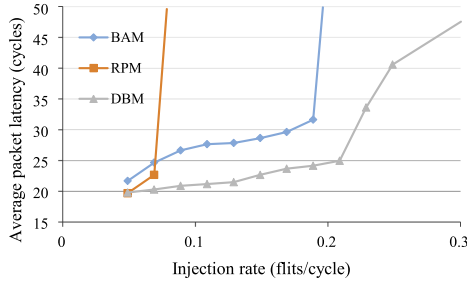


Fig. 5. Average packet latency in transpose traffic

The simulation in bit rotation pattern shows the similar result. The performance of RPM is the worst in three algorithms. It can be seen that in general, latency will increase as load (injection rate) increases. RPM is clearly not scalable as latency will dramatically increase with traffic load. Both DBM and BAM perform better than RPM. So, we only compare DBM with BAM in the following simulation.

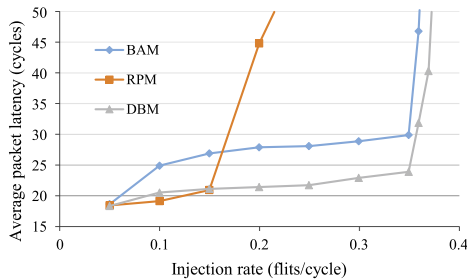


Fig. 6. Average packet latency in random permutation traffic

4.3 Impact of Buffer Size

The bit rotation traffic is considered to be the worst case for all the networks under study. We study network performance with different buffer under this traffic pattern.

The performance with different buffer sizes are showed in Fig. 8. DBM and BAM’s algorithm both improve given larger buffer sizes. Comparatively, DBM performance is more advantage. For example, relative to DBM, when the buffer size is 4, 6 and 8 respectively, DBM performance achieved 8.6%, 8.1% and 14.2% latency reduction and 6.3%, 8.8% and 12.2% throughput improvement.

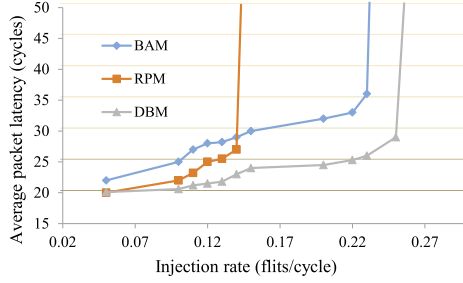


Fig. 7. Average packet latency in bit rotation traffic

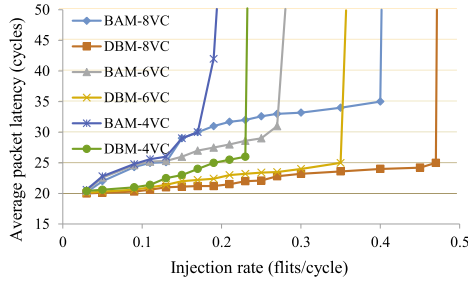


Fig. 8. Average packet latency with different buffer size

4.4 Scalability of Network Size

In 2D mesh networks, DBM exhibits better scalability than BAM with respect to network size. The performance with different network sizes are showed in Fig. 9. Using the uniform random pattern, we compare latencies by increasing the number of nodes in each dimension from 16 to 64. When the number of nodes from 16 to 64, relative to BAM, DBM achieved latency reduction from 13.8% to 18.1% and throughput improvement from 8.6% to 9.8%.

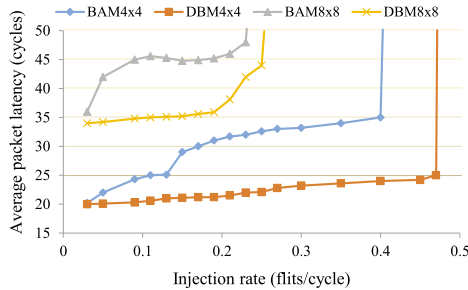


Fig. 9. Average packet latency with different network size

4.5 Discussion

DBM can improve the network performance by employing more restrictive flow control scheme than that in BAM’C the more remaining dimensional directions of replicated packet, the more buffer space in the next hop required. Although such bias may raise the chances of conflicts, it encourages packets to use more paths and enables DBM to unburden the network by balancing traffics.

In contrast, BAM in Duato’s framework, a packet is free to enter any queue as long as credit available. This freedom tends to form a locked ring in busy networks and requires the escape channels to break the deadlock which is very likely to end up with the “hot-spot” problem causing performance degradation.

More importantly, DBM promises the adaptively throughout the trip of a packet. However, in BAM, a packet has to enter into the DOR-based escape channel for deadlock avoidance when conflicts occur. As a consequence, within a typical injection rate range, traffic congestion tends to be alleviated in the networks using DBM. This explains why DBM is more adept at preventing the networks from performance degradation while the injection rate increases or traffic patterns become more adverse.

Understandable, a network’s overall performance is determined by the “worst case” routing. In fact, buffer size increment helps DBM attenuate the “worst case” impact, which happens when a queue becomes full. In other words, a larger buffer size makes a queue less likely to become full.

Specifically, the flow control DBMFC keeps the number of “bubbles” as balanced as possible in queues, and regardless of the buffer size, this mechanism remains functioning. However, this is not the case for BAM. The “worst case” in BAM is the “hot-spot” followed by DOR routing. Unfortunately, larger buffer size makes a queue capable of accommodating more packets implying that the queue contains more packets when it becomes full. Thus, more packets may enter into the DOR-based escape channel, which prolongs the average latency.

5 Conclusions and Future Work

In this paper, we design the DBM multicast routing strategy for 2D mesh networks. DBM algorithm provides a new way to implement high performance multicasting routing in interconnection networks. We prove that the proposed algorithm is deadlock-free while enabling minimal path and fully-adaptive multicast routing. Moreover, we complete some comparative work against RPM and BAM algorithms, which indicates that DBM can achieve more performance and scalability improvement under synthetic workloads. In the future work, we will study the micro-architecture of router based on DBM.

Acknowledgment. We thank the anonymous reviewers for their valuable feedback. We gratefully acknowledge members of Tianhe interconnect group at NUDT for many inspiring conversations. This project was partially supported by the National Key R&D Program of China under Grant No. 2018YFB0204300, and in part by NSFC. 61802416 and School Research Project of National University of Defense Technology under grants No. ZK20-18.

References

1. De Azevedo, M.M., Blough, D.M.: Fault-tolerant clock synchronization of large multicomputers via multistep interactive convergence. In: Proceedings of the 16th International Conference on Distributed Computing Systems, pp. 249–258 (1996)
2. Duato, J., Pinkston, T.M.: A general theory for deadlock-free adaptive routing using a mixed set of resources. *IEEE Trans. Parallel Distrib. Syst.* **12**(12), 1219–1235 (2001)
3. Furber, S., Galluppi, F., Temple, S., Plana, L.A.: The spinnaker project. *Proc. IEEE* **102**(5), 652–665 (2014)
4. Jerger, N.E., Peh, L., Lipasti, M.H.: Virtual circuit tree multicasting: a case for on-chip hardware multicast support. In: Proceedings of the International Symposium on Computer Architecture, pp. 229–240 (2008)
5. Jiang, N., et al.: A detailed and flexible cycle-accurate network-on-chip simulator. In: Proceedings of the IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), pp. 86–96 (2013)
6. Li, K., Schaefer, R.: A hypercube shared virtual memory system. In: Proceedings of the International Conference on Parallel Processing, pp. 125–132 (1989)
7. Ma, S., Jerger, N.E., Wang, Z.: Supporting efficient collective communication in NoCs. In: Proceedings of the IEEE International Symposium on High-Performance Computer Architecture, pp. 1–12 (2012)
8. Mckinley, P.K., Xu, H., Kalns, E.T., Ni, L.M.: Compass: efficient communication services for scalable architectures. In: Proceedings of the ACM/IEEE Conference on Supercomputing, pp. 478–487 (1992)
9. Navaridas, J., Lujan, M., Miguelalonso, J., Plana, L.A., Furber, S.: Understanding the interconnection network of spinnaker. In: Proceedings of the 23rd International Conference on Supercomputing, pp. 286–295 (2009)
10. Rodrigo, S., Flich, J., Duato, J., Hummel, M.D.: Efficient unicast and multicast support for CMPS. In: Proceedings of the 41st Annual IEEE/ACM International Symposium on Microarchitecture, pp. 364–375 (2008)
11. Wang, L., Jin, Y., Kim, H., Kim, E.J.: Recursive partitioning multicast: a bandwidth-efficient routing for networks-on-chip. In: Proceedings of the 3rd ACM/IEEE International Symposium on Networks-on-Chip, pp. 64–73 (2009)
12. Xiao, C., Yang, Y., Zhu, J.: A sufficient condition for deadlock-free adaptive routing in mesh networks. *IEEE Comput. Archit. Lett.* **14**(2), 111–114 (2015)
13. Xu, H., Mckinley, P.K., Ni, L.M.: Efficient implementation of barrier synchronization in wormhole-routed hypercube multicomputer's. *J. Parallel Distrib. Comput.* **16**(2), 172–184 (1992)