Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 93 (2016) 283 – 291

Procedia
Computer Science

6th International Conference On Advances In Computing & Communications, ICACC 2016, 6-8 September 2016, Cochin, India

A Novel Energy Efficient Multicasting Approach For Mesh NoCs

M.R. Arun[a,*], P.A. Jisha[a], John Jose[b]

[a]Mar Athanasius College of Engineering, Kothamangalam, India
[b]Indian Institute of Technology, Guwahati, India

Abstract

Network on Chip (NoC) is a scalable and flexible communication infrastructure which replaces dedicated point to point wiring between cores on a chip. Communication among the cores is established via routers interfaced to each core. Efficiency of NoC router plays an important role in determining the performance of the underlying network in multicore systems. In Chip Multiprocessors (CMP), cache coherence protocols, and cache miss reply from memory controller uses multicast communication. In early days, multicast communication is achieved by generating multiple unicast packets. The multicast message efficency can be improved by providing support at the hardware level. Multicast communication can be optimised by grouping destination cores of a multicast message into various partitions. Existing state of the art multicasting techniques on NoC issue one packet each to each destination partition, there by avoiding need for multiple unicast packets. In this paper, we propose an efficient multicast approach for 2D Mesh NoCs where a single packet issued from the source core, move through network as much as possible and then create duplicate packet as and when required. Experimental results show that our technique significantly reduces average multicast transaction latency and number of link traversals required to realise a multicast communication. This reduction in latency and link activity gives significant energy savings associated with multicast communication.

Peer-review under responsibility of the Organizing Committee of ICACC 2016

Keywords: On chip router; coherence message; Latency; link activity

1. Introduction

Rapid developments in VLSI technology allowed billions of transistors to be placed on a single chip which led to the integration of multiple IP cores on a single integrated circuit forming multicore systems. A dedicated, fast, energy efficient and reliable communication infrastructure is needed to meet the communication demands of such systems. Such multicore systems widely adopt Network on Chip (NoC)[1] framework for meeting their on-chip communication requirements. NoC replaces dedicated point-to-point bus based communication between cores by employing a grid of routers connected via communication links. Figure 1 shows a 9-core tiled chip organised as 3×3 mesh topology. Each core consists of an out-of-order executing superscalar processor, a private L1 cache and a slice of shared distributed

---

[*] Corresponding author. Tel.: +91-940-071-2789 ; fax: +91-485-225-6789.
E-mail address: arun.em6@gmail.com, jishaanil@gmail.com, johnjose@iitg.ernet.in

Peer-review under responsibility of the Organizing Committee of ICACC 2016
doi:10.1016/j.procs.2016.07.212

L2 cache. Each core is connected to a switching element called a NoC router. Each router (except routers at edge / corner) is in turn connected to four neighbors in the East, West, North and South directions. The cores are arranged as rectangular tiles on the chip and are interconnected using a network of routers. Cores communicate with each other by exchanging packets. The source core (core that has to send data) creates a packet with necessary control information and injects it to the router connected to it. The control information (header) contains the source address, destination address, packet number, packet type, etc which are required for forwarding the packet at various intermediate routers. The payload of the packet contains the data to be communicated.

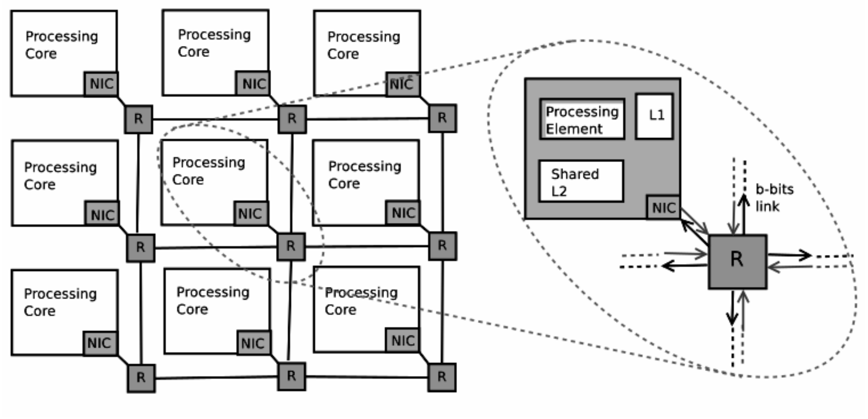

Fig. 1. A 2D mesh topology showing processing core and NoC router interaction.

The Figure 2 shows a NoC router architecture with input buffers and virtual channels[2]. The multicast controller block is the only additional functional unit that we have added for the proposed technique. Inter-router packet channels from four different directions and local processing core terminates at the virtual channels (input buffer) of NoC router. The routing logic extracts the destination address of the packet and determine the next outgoing channel. It also searches for a free virtual channel (VC) in the next downstream router. Virtual Channel allocator assigns a VC for every incoming packet that have completed routing. If multiple packets compete for same outgoing port, the switch allocator will resolve the conflict. A conventional input buffered router has a two stage pipeline of which routing is done in stage one, virtual channel allocation and switch allocation are done in stage two. It takes one cycle for a packet to traverse through a crossbar followed by movement through the inter-router channel to reach the next downstream router. The performance and power consumption of a NoC is greatly dependent on the efficiency of the router architecture and routing mechanism. Modern applications generate a lot of multicast traffic. Multicast message (one-to-many) is defined as a data that has a single source core but multiple destination cores. A special mechanism is needed to ensure timely and efficient delivery of multicast messages. Currently multicasting is handled by sending multiple unicast messages. This approach is ineffective as the percentage of multicast traffic to total network traffic increases beyond 2% percent[8]. Few work has been done[5], which provide additional facilities at the NoC router level to handle multicast messages.

The rest of the paper is organized as follows: Section 2 reviews related work and motivation for the proposed work. The proposed method is discussed in Section 3. We discuss experimental analysis in Section 4 and concludes our work in Section 5.

2. Related Work

Hardware level modification[7] at the NoC router level for implementing multicast schemes can be broadly classified into path-based and tree-based approaches[9, 10]. A spanning tree is built at the source node and a multicast packet is send down the tree in tree based approach. The root of the tree will be source while leaves of this tree are destinations.

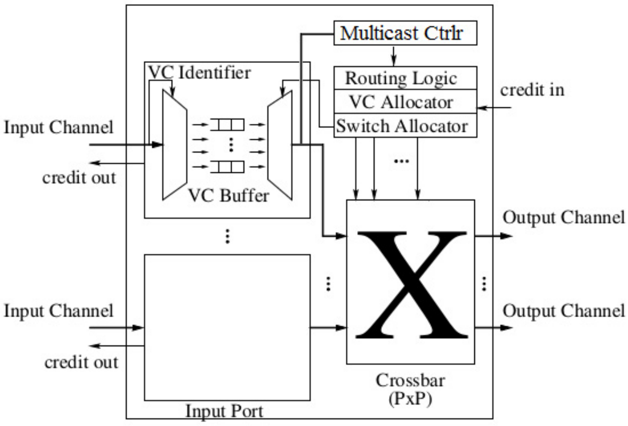

Fig. 2. Conventional NoC router architecture with an additional Multicast Controller for supporting our proposed MDND.

The packet is replicated along its path at every child node so as to reach all nodes of the tree[15]. In the path-based multicast method, the router prepares a packet for delivering to a set of destinations by placing the list of destinations in the packet itself. At each intermediate destination a copy of the multicast packet is delivered to local core and the packet is forwarded to next destination. In this way, the message is eventually delivered to all the specified destinations. A number of studies have shown that path-based methods exhibit superior performance characteristics over tree-based counterparts[10]. The path-based approach does not replicate messages within the network, and hence they will not increase message contention. However it takes long time for the multicast message to reach all destinations. This increase multicast transaction latency. Multicast transaction latency is defined as the total time (cycles) elapsed between generation of multicast message and reception of this multicast message at the farthest destination from source. To reduce the transaction latency of the multicast message, destinations can be divided into several disjoint destination subsets at the network interface of the source router. The message is send via several separate packets to each of these destination sets[15]. In this paper, we present a novel approach to determine the point at which the duplicate packets are to be generated. We keep an optimal balance to reducing average multicast transaction latency as well as the number of link traversals required for completing the multicast transaction.

Due to the fact that the multicast communication is used commonly in various parallel applications, there have been several attempts to improve the performance of multicast communication in 2D NoCs[7]. Virtual Circuit Tree Multicasting (VCTM)[11], Recursive Partitioning Multicast (RPM)[12], and Hamiltonian path-based multicast algorithm for NoCs[13, 14] are three recent works focused on 2D NoCs. VCTM and RPM are tree-based methods and few algorithms[13, 14] explores path-based methods. In VCTM method, when the number of destinations are more, a large number of setup messages must be delivered into the network (before delivering the real multicast message) which decreases the performance significantly. The area over-head of VCTM is relatively high due to the table that store the information of a virtual circuit tree at each router. In RPM method, the processing of the header information is complex. Another disadvantage of VCTM and RPM method is that a message may hold several channels for extended periods of time to receive all requested output channels, thereby increasing network contention[15]. A solution to overcome the disadvantages of tree-based multicast is to utilize path-based multicast routing. One of the recent work[5] has mentioned about a multicasting approach which can improve performance by recursive partitioning. We compare our results with this technique also.

3. The proposed router architecture

To overcome the limitations of the state of art multicasting approach we propose a new router architecture that has an additional hardware support for efficiently handling multicast messages. Multicasting is achieved by generating duplicate packets at various intermediate routers so that the multicast message reaches all the intended destinations

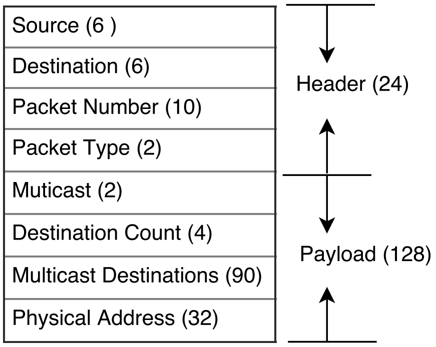

Fig. 3. Packet structure.

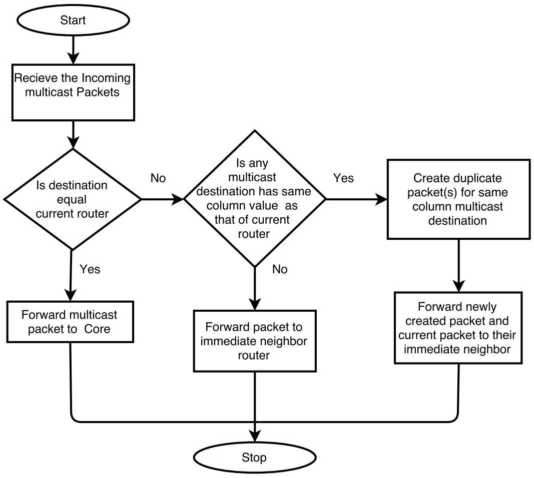

Fig. 4. Working of multicast controller in the proposed NoC router.

in the shortest possible time. In our method, we use a novel approach that chooses the point of packet duplication carefully such that number of duplicate packets created, overall multicast transaction latency and average number of hops taken by various packets for completing multicast transaction are minimized. In our proposed technique the message duplication is not happening at partition level. We generate at most four initial multicast packets from the source router (1) Horizontally along East direction (2) Horizontally along West direction (3) Vertically along the North direction (4) Along South direction depending on the position of destination routers. Intermediate routers generate duplicate packets from these initial four packets. Hence the point of duplication can be non-destination router also. So our technique is called Message Duplication in Non Destination (MDND). We use this convention in all subsequent references in this paper.

The additional unit of multicast controller in Figure 2, is the extra circuitry that we have added on the conventional router for providing MDND support. The multicast controller identifies multicast packet, extracts destination address and provide packet duplication if needed. Multicast message is typically a single flit packet with few bits reserved for control information. The details regarding the address of multicast destinations is stored in the packet payload. We consider a flit structure shown in Figure 3 for a NoC system that support MDND. The flit has 128 bit payload field and 24 bit header field, there by making 152 bit flit channel. We consider an 8×8 mesh topology with 64 cores for all our experimental studies. Each of the 64 core is uniquely addressed by 6 bits. So our header consists of 6 bit source, 6 bit destination, a 10 bit cyclic packet number and a 2 bit packet type. A special bit combination in the packet type identifies a packet as multicast packet. In payload field, 2 bit multicast field indicates whether the operation is a cache update or cache invalidate or any other cache coherence control message. We can support up to 15 destinations (4 bit destination count) using this multicast packet structure. The multicast destination field in the packet payload contain the 6 bit address of at most 15 destinations. A 32 bit physical address field indicate the address of memory location for which the coherence protocol message is issued.

For routing packets, we used dimension order x-y routing which is deadlock and live lock free.

3.1. The Concept of MDND

Once a processing core has decided to generate a multicast message, it forwards this information with the set of multicast destinations to the local router. The multicast controller of the local router creates multicast packets with the structure shown in Figure 3. The multicast controller then divide the entire mesh topology into four zones.
(a) ***East zone:*** The multicast destinations that has a column value larger than the source router's column value.
(b) ***West zone:*** The multicast destinations that has a column value smaller than the source router's column value.

(c) ***North zone:*** The multicast destinations that has a row value larger than the source router's row value and same column value as of source router.
(d) ***South zone:*** The multicast destinations that has a row value smaller than the source router's row value and same column value as of source router.
For every zone containing at least one multicast destination, an initial packet is generated with destination as the farthest multicast destination in that zone with respect to source core. We call these packets as *Eastern, Western, Northern* and *Southern* packets respectively. The multicast payload field of the initial packets contains the details of the destinations belonging to that zone. The initial packets then traverse in the appropriate zone. Note that the *Eastern* and *Western* packet may take few X movements followed by Y movements depending on the location of the farthest destination in respective zones. But Northern and Southern packets have only Y movements as the zone is limited to one column

All Eastern and Western packets upon reaching each intermediate router checks whether there is a multicast destination in the same column. There can be three possible outcomes.
Case 1: If intermediate router itself is a multicast destination, then a special duplicate packet is created by multicast controller and forwards it to the local destination core.
Case 2: If there are muticast destinations in the same column as that of intermediate router either in Northern side or in Southern side or both, then multicast controller will create duplicate messages from that intermediate router with payload field containing the multicast destinations in the column. So there will be at most two duplicate packets created; one for North direction and another for South direction.
Case 3: If there is no multicast destination in the column, duplicate packets is not created.
In all of the above cases, original *Eastern* / *Western* packet will proceed to the next neighbor as per XY routing. Duplicate packets created from *Eastern* and *Western* packet will move in North or/and South direction till the farthest destination in the same column. These packets on each intermediate router checks whether the local core is a multicast destination or not, if so a special duplicate packet is created by multicast controller and forwards it to the local core. The same will be the operation for initial *Southern* and *Northern* packets. Algorithm 1 and 2 list out the steps taken at source router and intermediate router respectively. Figure 4 represents a detailed flowchart of the same.

Alg 1 Steps for creating initial packets at source router

Step 1: Divide the multicast destination in the packet M_d in to four zones, namely *Eastern*, with destination having larger column value than source router, *Western* with destination having lower column value than source router, *Northern* with destination having higher row value and same column value as of source router *Southern* with destination having lower row value and same column value as of source router.
Step 2: Create initial packets with destination as the farthest destination in each of the Eastern, Western, Northern and Southern zones and the rest multicast destinations in each zone as multicast destination in payload correspondingly.
Step 3: Inject the initial packet(s) into network.

Alg 2 Steps for duplicating a flit at intermediate router

Step 1: If destination is not current router, find multicast destinations for same column as of current router. Then those multicast destinations are divided into 3 zones, *Northern* with row value more than current router, *Southern* with row value less than current router, *Self* with same row and column value.
Step 2: Create duplicate packets for *Northern, Southern, Self* with destination as the farthest multicast destination the respective direction.
Step 3: Inject the duplicate packet(s) into network.

3.2. Illustration

We will illustrate the MDND approach with an example. Consider an 8×8 mesh with 64 cores (0,1...63) as shown in Figure 5. Consider a multicast message to be sent from source core 27. Let us assume there are 8 multicast destination

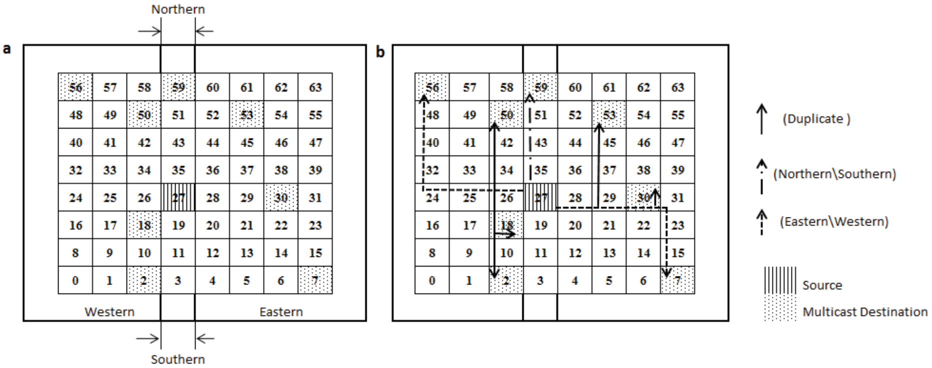

Fig. 5. (a) Various zones with respect to source 27; (b) Initial and duplicate packet movements in NoC for realising multicast message having source at 27 and destination M_d={2,7,18,30,50,53,56,59}.

represents by M_d={2,7,18,30,50,53,56,59}. This information is sent from source core 27 to its router. The multicast controller at 27 examines the M_d list and finds that there are multicast destination in *East, West* and *North* zone. From Figure 5, we can see that there is no multicast destination to the south zone of 27. As per XY, routing 7 is the farthest multicast destination in *East* zone and there are 2 more multicast destination (30 and 53) in that zone. So *Eastern* packet from 27 contain 30, 53 in the payload (in destination address field) and destination count field as two. This can be represented as *27_East*{ *7, (30, 53)*}. Similarly we get *27_West* {*56, (2, 18, 50)*} and *27_North*{*59, ( )*}. These three packets are forwarded from 27 to the appropriate neighbors. *27_East* upon reaching 28 finds no multicast destination in the column, so it moves to router 29 as per XY routing. At 29, we can see that 53 is a multicast destination in the same column. So a duplicate packet to 53 is created. One more duplicate packet is created when *27_East* reaches 30. This duplicate is forwarded to the local core of 30. Eventually the *27_East* reaches 7 via 31, 23 and 15. Similarly *27_West* packet upon reaching 26 will create 2 duplicate packets, one to destination 50 and other to destination 2 with payload 18 as multicast destination. *27_West* then move to 56 via 25, 24, 32, 40 and 48 without any further creation of duplicate packets. Note that all duplicate packet created in this illustrative example have source address field in the packet header as 27. This is to help the multicast destinations in identifying the actual source of multicast message.

4. Experimental Analysis

We now compare the performance of MDND with SMDP[5] and UbM[7] techniques proposed in recent the past. In SMDP technique, based on the location of source core, rest of the routers are partitioned into eight zones namely North East, East, South East, South, South West, West, North West and North. SMDP is a path based technique where the unicast packets are send to one of the destinations in each partition. Upon receiving a multicast packet in a partition, this partition is technique applied recursively until all destinations are covered. UbM is the conventional method of sending multicast message by multiple unicast packets from source core to each individual destinations.

4.1. Experimental Setup

We use Booksim[16], a cycle accurate network simulator, that models a two-cycle router micro-architecture in sufficient detail. The simulator is modified to incorporate UbM, SMDP and MDND multicasting techniques. We evaluate our proposed technique using three standard synthetic traffic patterns; uniform, tornado and bit-complement 8 × 8 mesh networks. We consider five VCs per input port and single flit packet for conducting experiments. Percentage of multicast traffic is varied from 1%, 2% and 4% of general traffic. We evaluate average multicast latency, multicast link traversal count, multicast transaction latency, and compare the performance of proposed technique with baseline architecture[5]. We also evaluate performance of our model under multicast traffic generated by multi-threaded benchmarks running along with 64 applications taken from SPEC2006 CPU benchmark.

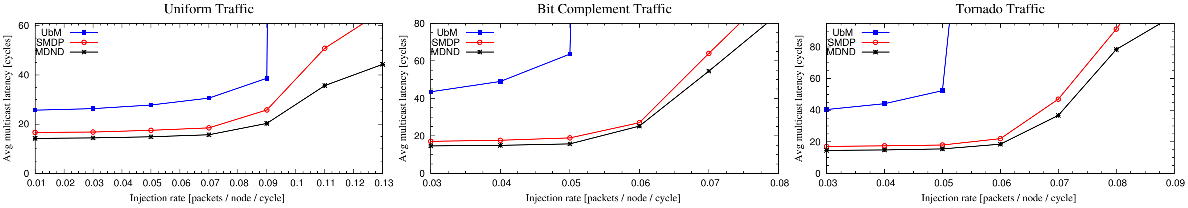

Fig. 6. Analysis of average multicast packet latency at 4% multicast traffic.

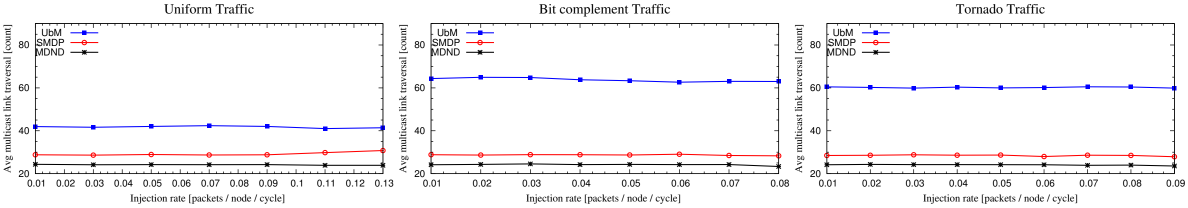

Fig. 7. Analysis of average multicast link traversal count at 4% multicast traffic.

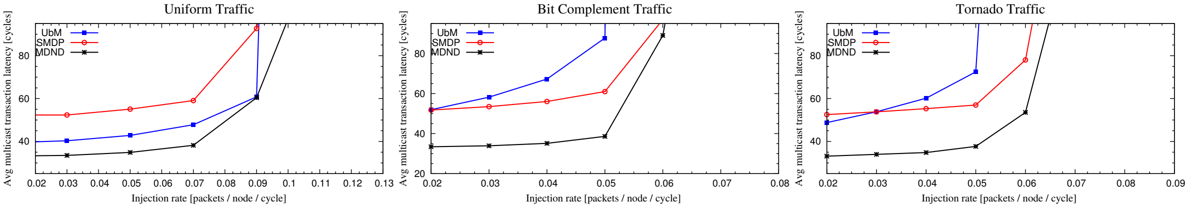

Fig. 8. Analysis of average multicast transaction latency at 4% multicast traffic.

4.2. Evaluation of Average Multicast Packet Latency

Figure 6 contains the injection rate vs average multicast packet latency for various synthetic traffic patterns under 4 % multicast traffic. We modified Booksim in such a way that one multicast message is issued in every 25 regular packets. For every individual packet generated for realising a given multicast message we collect latency and compute the average. We can see that MDND achieve significant reduction in average multicast packet latency with respect to SMDP and UbM technique right from zero load to saturation. We can see that the saturation injection rate is also significantly extended by using MDND technique. This is because of the reduction in number of packets needed to realize a multicast transaction by using MDND technique. We also observe that majority of all duplicate packets are travelling in a single column. The initial four packets (at most) forwarded to four different zones adopts a divide and conquer approach to cover all destination in minimum time.

4.3. Evaluation of Multicast Link Traversal Count

Multicast link traversal count is a measure of the total number of hops taken through the NoC by various packets of a multicast message. This count is affected by (1) number of packets generated for a multicast message (2) number of links each of these packets have traversed. Figure 7 shows injection rate vs average multicast link traversal count. We can see that for various synthetic traffic patterns under 4 % multicast traffic rate, our technique has the minimum number of link traversal to realise a multicast transaction. Proposed technique substantially reduces both number of packets and hops taken per packet to deliver a multicast message. This indirectly contributes to reduction of dynamic power dissipation in NoC. Our power simulation on Orion2.0[6] shows that when compared to UbM, SMDP achieve 28% of reduction in link energy where as MDND achieve 40% of reduction in link energy for handling multicast traffic alone.

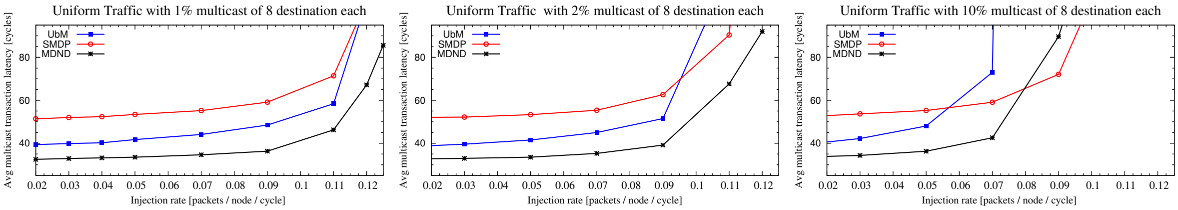

Fig. 9. Analysis of average multicast transaction latency under different multicast traffic rates.

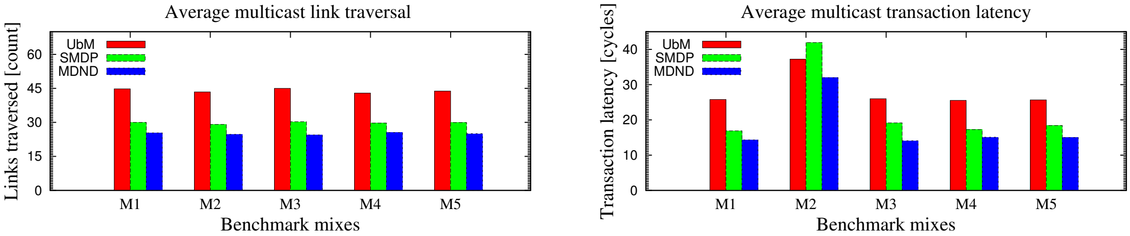

Fig. 10. Analysis of various multicasting techniques using real workload traces of varying network injection rates. For each cache miss packet we consider 8 flit reply packet.

4.4. Evaluation of Multicast Transaction Latency

Multicast transaction latency is defined as the total time (cycles) elapsed between generation of multicast message and reception of this multicast message at the farthest destination from source. Indirectly it is the time required to spread a multicast message from source to all its multicast destinations. Reduction in multicast transaction latency is very much needed for a multi-threaded application running in source core to continue its execution. Figure 8 shows the plot for injection rate vs average multicast transaction latency. We can see that for various synthetic traffic patterns under 4 % multicast traffic rate, our technique has significant reduction in average multicast transaction latency with respect to SMDP and UbM right from zero load to saturation load. We can see that the saturation injection rate is also significantly extended by using MDND technique. This will help coherence transactions to complete cache invalidation requests in a short time.

4.5. Evaluation of Performance at Varying Multicast Rate

We conduct analysis at 1%, 2% and 10 % multicast traffic for various synthetic traffic patterns. Figure 9 shows injection rate vs multicast transaction latency at different multicast traffic rates for uniform pattern in 8 × 8 mesh network. We observe that irrespective of traffic rate our technique is performing well. This makes our technique best suited for multi-threaded application with non-uniform cache coherence traffic rate. All our previous results (Sections 4.2-4.4) are for 4% multicast traffic.

4.6. Evaluation of Performance Under Real workloads

Apart from analysing synthetic traffic patterns, we study the effect of our proposed router under real traffic also. We generate network events (cache misses) from a 64 core full system simulator that runs one SPEC 2006 CPU benchmark per core. We incorporate coherence messages generated by PARSEC and SPLASH work load (multithreaded applications) on the same 64 core parallelly along with the SPEC 2006 multiprogrammed workload. This network trace is fed to our Booksim module to evaluate the performance of various multicasting techniques. We classify the work load into 5 different mixes (M_1, M_2..M_5) based on the cache miss rates of application running on the cores. (M_1 = 100% Low MPKI application, M_2 = 100% Medium MPKI application, M_3 = 100% High MPKI application, M_4 = 50% Low and 50 % Medium MPKI application and M_5 = 50% Low and 50 % High MPKI application.) Figure 10 shows the comparison of multicast link traversal count and average maximum transaction latency for various mixes.

We can see that across all mixes, MDND achieves lowest multicast link traversal count (hence lower dynamic link power dissipation) and lower multicast transaction latency. Thus we prove that our technique is better under real work load also.

5. Conclusion

An energy efficient router architecture for multicasting is proposed. Our model limits the link traversals to an optimum value to achieve faster delivery of multicast message to all destinations. This reduced both dynamic power consumption and multicast transaction latency. The light weight multicast controller logic in the proposed router ensured that the power and area overhead in the proposed design is negligible compared to the latency and link traversals reductions achieved. In larger networks, across all traffic patterns examined, MDND has the least multicast transaction latency, average multicast packet latency and average multicast link traversals in pre-saturation loads. Hence we propose that have MDND multicast facilitation is a superior design choice for NoCs that carry large share of multicast messages.

Acknowledgements

Authors like to thank the Computer Science and Engineering Department of Indian Institute of Technology Guwahati, for providing facility to carry out advanced experiments as part of the work.

References

1. W Dally and B Towles. Principles and Practices of Interconnection Networks. Morgan Kaufmann Publishers Inc.,2003, San Francisco, CA, USA.
2. W. J. Dally. Virtual-channel flow control. Proceedings. 17th Annual International Symposium on Computer Architecture, Seattle, WA, 1990, pp. 60-68.
3. X. Lin and L.M. Ni. Multicast Communication in Multicomputer Networks, IEEE Trans. Parallel and Distributed Systems, no. 10, pp. 1105-1117, 1993.
4. E. Carara and F.G. Moraes. Deadlock-Free Multicast Routing Algorithm for Wormhole-Switched Mesh Networks-on-Chip, Proceedings IEEE CS Ann. Symp.VLSI (ISVLSI), pp. 341-346.
5. M R Arun and P A Jisha. SMDP- Single Message Duplicate in Partition, A Multicast routing method in mesh 2D NoC, Proceedings 3rd International Conference on Innovations in Information Embedded and Communication Systems 2016, pp 676 - 680.
6. A. B. Kahng and et.al, ORION 2.0: A fast and accurate NoC power and area model for early-stage design space exploration, DATE, 2009, pp. 423-429.
7. M. Ebrahimi and et.al, Path-Based Partitioning Methods for 3D Networks-on-Chip with Minimal Adaptive Routing, in IEEE Transactions on Computers, vol. 63, no. 3, pp. 718-733, 2014.
8. Abad and et.al, MRR: Enabling Fully Adaptive Multicast Routing for CMP Interconnection Networks, Proceedings IEEE 15th International Symposium. High Performance Computer Architecture (HPCA), pp. 355-366, 2009.
9. R.V. Boppana and et.al, Resource Deadlock and Performance of Wormhole Multicast Routing Algorithms. IEEE Trans. Parallel and Distributed Systems. vol. 9, no. 6, pp. 535-549, 1998.
10. D. Panda and et.al, Multi Destination Message Passing in Wormhole K-Ary N-Cube Networks with Base Routing Conformed Paths, IEEE Transactions on Parallel and Distributed Systems, pp. 76-96, 1999.
11. N.E. Jerger and et.al, Virtual Circuit Tree Multicasting: A Case for On-Chip Hardware Multicast Support, Proceedings 35th International Symposium. Computer Architecture (ISCA), pp. 229-240, 2008.
12. L. Wang, H. Kim and E.J. Kim. Recursive Partitioning Multicast: A Bandwidth-Efficient routing for Networks-on-Chip. Proceedings International Symposium Networks-on-Chip (NOCS), pp. 64-73, 2009.
13. E. Carara and F.G. Moraes. Deadlock-Free Multicast Routing Algorithm for Wormhole-Switched Mesh Networks-on-Chip, Proceedings IEEE Computer Society Annual Symposium on VLSI,pp. 341-346, 2008.
14. M. Daneshtalab. A Generic Adaptive Path-Based Routing Method for MPSoCs. Elsevier Journal Systems Architecture. vol. 57. no. 1, pp. 109-120, 2011.
15. J. Duato and et.al, Interconnection Networks: An Engineering Approach, 2003.
16. Nan Jiang and et.al, A Detailed and Flexible Cycle-Accurate Network-on-Chip Simulator. Proceedings of IEEE International Symposium on Performance Analysis of Systems and Software,pp. 86-96, 2013.