International Journal of Computer Information Systems and Industrial Management Applications.
ISSN 2150-7988 Volume 9 (2017) pp. 087–095

Deflection Router for Mesh NoC with Multicast Support Mechanism

Elizabeth Isaac[*1], **M Rajasekhara Babu**[**1] **and John Jose**[2]

[1]School of Computer Science and Engineering
VIT University Tamil Nadu,India
[*] *elizabeth.issaac@gmail.com*

[**] *mrajasekharababu@vit.ac.in*

[2]Department of Computer Science and Engineering,
Indian Institute of Technology Guwahati, Assam, India
johnjose@iitg.ernet.in

Abstract: **Network on Chip (NoC) has been introduced as a cost effective solution to address the on chip design challenges of the dedicated bus-based communications for multicore Systems-on-Chip(SoC). The increase in core density for a multicore system and parallel execution of programs over these cores contribute to the multicasting. Multicast communication results in generation of multiple packets from a single source. Routers with input buffers form the backbone of a traditional NoC based communication system. Buffer-less NoCs are gaining popularity due to simplicity in the router design, low power consumption, and less chip area. Considering the cost overhead of the buffer NoCs deflection routers with minimal number of buffers are gaining importance. All architectural enhancement proposed in NoC systems are focusing in input buffered routers. We propose a novel cost effective deflection architecture that facilitates multicast support. We are making use of a partitioning mechanism for the flit duplication. Experimental analysis proves that our technique substantially reduces average transaction latency of multicast packets and link traversal count without increasing the average deflection rate.**

Keywords: bufferless, deflection routing, link traversals, multithreaded, packet duplication, transaction latency.

I. Introduction

Recent years, due to the advancement in the VLSI technology, the number of processing cores that can be integrated on a single chip increased rapidly. This swift transition in a multi-core chip paves way for running multithreaded applications as well as multiprogrammed workloads. The multithreaded applications share some common data that are distributed across multiple cores. To handle the information exchange among these cores, multi-core processors need an efficient on-chip communication framework. Network on Chip (NoC) is introduced as an efficient and cost effective solution to connect all the cores. NoC is a flexible and scalable communication subsystem on an integrated circuit, typically between the cores in a SoC (System on Chip).

A typical NoC based system consists of an interconnecticon of the cores through a set of routers. Each core consist of an out-of-order superscalar processor with a dedicated L1-cache and a distributed shared L2-cache. Cores communicate each other to service cache misses and coherence transactions, in the form of packets. A flit is a basic unit of a flow control between a pair of routers. Wormhole routing [9],[10] based on credit or handshaking flow control assure streamlined flow of flits through the routers.

Traditional input buffered NoC have buffers at each input port. Flits reaching the buffers can reside in the buffers until a productive output port is obtained. The buffers are power hungry and the buffer management circuits are complex [23]. Hence the buffer-less routers gained popularity than the traditional buffered NoC. Eliminating the buffers from the router design simplifies the router design and cuts down the area and power consumption. Packets reside in the bufferless router for a definite number of clock cycles in the pipeline latches, buffer-less router should forward all the incoming packets to the adjacent router. Under such circumstance few packets may be deflected away through non-productive port. But buffer-less NoCs incur significant performance degradation under heavy traffic. This is due to high deflection rate and higher number of port conflicts. In a bufferless system, contention of two flits for the same output port will force one flit to choose the available port, making other flit to deflect to a non-productive port. Deflected flit eventually reach the destination by proper livelock prevention mechanisms. Thus deflection routing emerges as the most popular choice of routing technique in buffer-less network [12],[13],[14].

Multicast messages are generated when the working set of the physical memory address are distributed across different cores. A highly multithreaded application, generates many multicast messages. One-to-many communication is frequently arising in multi-threaded parallel execution

environments due to synchronization of applications incoming in multiple cores, cache invalidation messages generated during cache coherence implementation and clock synchronization. Cache-coherence [16] is a mechanism used to maintain consistency among the shared data kept in the caches. Either write-invalidate or write update protocols can be employed to ensure the coherence among the multiple copies of the same data. Invalidation is done by propagating a multicast message to every core where the duplicate data resides. Cache invalidation [17], [18] techniques are employed to keep the data consistent across the caches. The applications in the cores can continue their execution only after successful deliver of invalidation message to respective sharer cores. Such multicast message over a conventional NoC framework can experience a significant delay and power consumption [3].

The conventional NoC router framework does not employ any unique hardware for handling multicasting. Slight modifications in the router architecture can help multicast communication. Even though the multicast packets can be serviced just like the unicast packets, this technique suffers from additional resource utilization and long waiting time due to the serialized nature messages. This is due to the decomposition of the message to serve each unique destination independently. Traditional unicasting approach cannot tackle such high sporadic traffic generated by multithreaded applications. But a multicast support at hardware level will improve the network performance in a power efficient way. Our contributions are:

- A new NoC router design for multicasting which works by deflection routing with minimal buffering support. This multicast router employs deflection routing but can use small buffer space for enhancing the performance.
- A new multicasting approach for minimally buffered NoC that takes into consideration the congestion parameter for the duplication of the packets. This approach divides the routers into different regions and duplicates are generated for each region.
- The evaluation of the proposed multicast approach is compared with the Recursive Partitioning, Hamiltonian Based and Unicast as multicast techniques.

In this paper, we propose a congestion aware, region based multicast support mechanism for buffer-less NoCs. Rest of the paper is organized as follows. The related works are covered in Section II. The motivation of this work and background of network partitioning in buffer-less routing are illustrated in Section III and IV respectively. The proposed work is introduced in Section V. Implementation details and experimental results are given in Section VI and we conclude our work in Section VII.

II. RELATED WORKS

Buffer-less NoCs are relatively new and promising area which work on the principle of the deflection routing. But unfortunately no multicast support is used for conventional bufferless NoC routers. A fine turning in the router architecture is inevitable to handle multicasting in buffer-less NoCs in a better way.

Several works have been proposed, to modify the traditional input buffered router microarchitecture, to facilitate multicast operations. One naive method is to disintegrate multicast message into the multiple-unicast packets. Each of the unicast packet is routed independently over the network to its respective addresses. This scheme increases the network traffic which leads to high message latency.

Hardware based multicast support systems can be broadly divided as path-based and tree-based [4], [5], [6] techniques. These two methods differ mainly the way the messages are routed to various multicast destination through the network. In path based multicasting, the duplication happens only at destination routers while traversing through every destination in a sequential way. Eventually the flit gets delivered at all the specified destination but takes long time to service all the destinations. This scheme does not increase any network congestion as there is no packet duplication. This scheme suffers from high 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 to all its destinations. To overcome the disadvantage of high transaction latency, destination list is divided into disjoint set. The message through duplication of the flits are sent to each of these destination list. These techniques are grouped as tree based, where the multicast message replication happens at any intermediate router not necessarily belongs to destination vector.

In path based approach, destination nodes are ordered at the source to find the minimal path. This results in increase in the overall latency due to the overhead at the source node. The message gets delivered at its last position without replication, after servicing all the elements in the destination vector.

Routers in 2D mesh are represented by Cartesian coordinate system which indicates the position along X and Y directions. In Hamiltonian path based routing strategy, each router is designated with a unique id starting from 0 to N-1 [8]. The naming starts from first router (0,0) then continuously assigning labels from left to right for even rows and right to left for odd rows. Multi-path routing algorithm is an extension of Hamiltonian routing strategies to reduce the path length in the network. The set of destination nodes is partitioned into two subsets namely left and right, whose ID are greater and smaller or vice versa than that of source, respectively.

In XY path based multicasting [19], a packet is sent along the X-direction and then propagated along the Y-channel to reach the first destination. After that, the message is forwarded to its subsequent destinations. Overhead channel dependencies [21] may arise due to the dispatch of similar message to many destinations.

In column path multicast routing [22], the destination

node set is partitioned into at most 2k subsets, where k is the number of columns. A packet is sent along the row direction and duplicated for every destination along the column. A single copy of the message is sufficient to service the destinations, if they occur either above or below a node along the row.

In tree based multicasting, [2] the packet travels along a common path as far as possible and branches (replicate) into separate copies of message that gets transmitted to disjoint unique subset of destinations. In Virtual Circuit Tree-Based Multicasting (VCTM) [3], a unique VCT number identifies the source-destination pair. However, it uses a lookup table based multicasting router which has high power and area overhead.

The Recursive Partitioning Multicast(RPM) [7] makes use of the concept of region division to replicate the packets. Based on the source node, the network is divided into eight regions. Replication to any region takes place only if the destination falls in that particular region.

Two replication schemes are popularly adopted; synchronous and asynchronous replication [11]. Asynchronous replication is preferred over the synchronous replication because an independent packet replica will be formulated without an overhead of a feedback architecture.

III. Motivation

Multicasting techniques in NoC proposed so far is on input buffered. To the best of our understanding so far no work is proposed to facilitate multicasting in bufferless NoCs. There is a growing popularity for buffer-less deflection routers for large NoC Systems. Multicasting support at router level for such systems involves greater research interest.

These issues cannot be left unnoticed since it has series impact on the performance of chip multicore system. Hence it is desirable to introduce a new router architecture to enhance the system performance.

The main objective of our work is introduce a multicasting technique to reduce the transaction latency of the packets in a buffer-less situation. However, the afore mentioned multicasting methods (Section 2) employed by the buffered routers cannot be directly adopted for the buffer-less NoCs. It is not advisable to utilize these strategies as they impose serious storage overhead which is a critical bottleneck in bufferless NoCs. All the intended destination can be reached effectively only if we generate required number of duplicates. This duplication can affect the performance of the NoC systems due to the storage bottleneck in buffer-less NoCs and also due to the added traffic. So, in order to implement multicasting in a buffer-less NoC, we need to have replication restriction which can avoid flooding of the packets. This situation has motivated us to put forward a new multicast hardware scenario that support in a buffer-less framework.

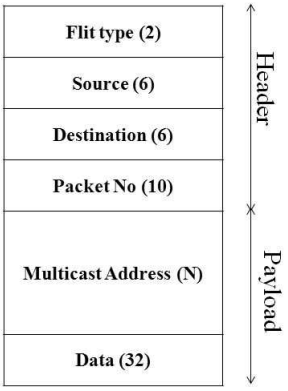

Figure. 1: Packet Structure

IV. Buffer-less Multicast Routing

The NoC system under consideration is a 2D mesh organized on Nx N. Each node is represented by (X_{node}, Y_{node}) making use of a Cartesian Coordinate system which indicates the position along the X and Y direction. Each node is given a label starting from 0, and label $Node_{ID}$ is given by,

$$Node_{ID} = (X_{node} * N) + Y_{node} \quad (1)$$

For each node position (X_{node}, Y_{node}), the entire network is partitioned into 4-subregions with respect to the source node (X_{source}, Y_{source}) based on the following conditions.
$Region_{NE}$: with nodes (X_{node}, Y_{node}) such that $(X_{node} > X_{source})$ and $(Y_{node} \geq Y_{source})$
$Region_{SE}$: with nodes (X_{node}, Y_{node}) such that $(X_{node} \geq X_{source})$ and $(Y_{node} < Y_{source})$
$Region_{NW}$: with nodes (X_{node}, Y_{node}) such that $(X_{node} \leq X_{source})$ and $(Y_{node} > Y_{source})$
$Region_{SW}$: with nodes (X_{node}, Y_{node}) such that $(X_{node} < X_{source})$ and $(Y_{node} \leq Y_{source})$

Message carries all the information needed for routing the data packets. For a multicast message the destination header list should contain all the designated destinations. In order to reduce the communication latency, the header information need to be minimized. This helps to increase the effective network bandwidth. Different multi-address encoding schemes [1] can be employed depending upon multicasting strategy. Several efforts are made to find out an effective way to manage the multiple destination addresses such as all-destination encoding, bit string encoding and multiple-region broadcast encoding. In order to reduce the header processing time in our work we are making use of the bit-string encoding. This scheme performs well even if the destination list is large.

Figure 1 shows the packet structure that is used to facilitate multicasting in buffer-less NoCs. Flit type specifies the type of the flit i.e *Head, Body or Tail*. In a 8x8 mesh network, each router is uniquely addressed by 6 bits. Hence we use 6-bits each for both source and destination. A 10-bit cyclic packet number is also a part of packet header which uniquely identifies the flits in the network. A 2-bit multicast field in the payload indicates that the packets contain multicast information. The multicast destinations

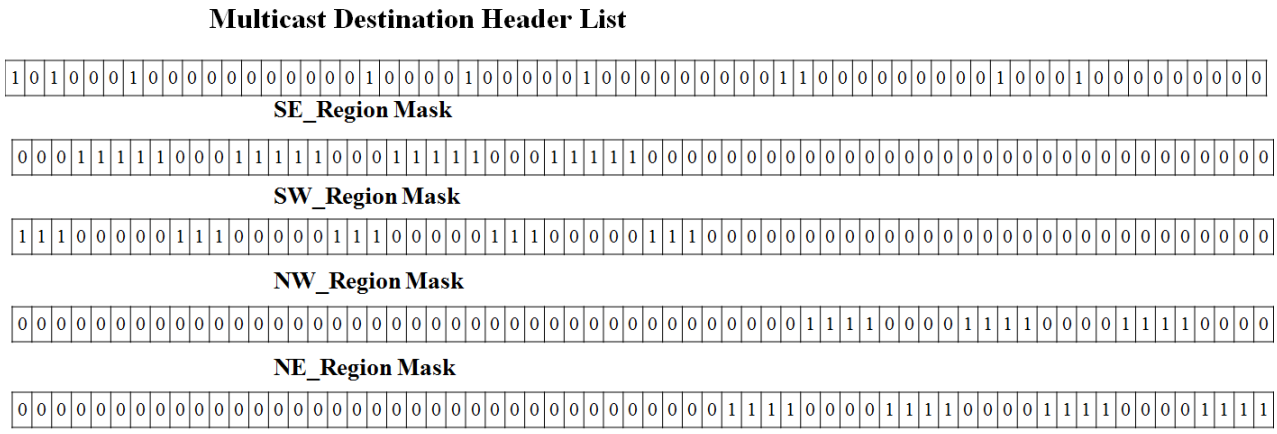

Figure. 2: Network partitioning based on source node position.

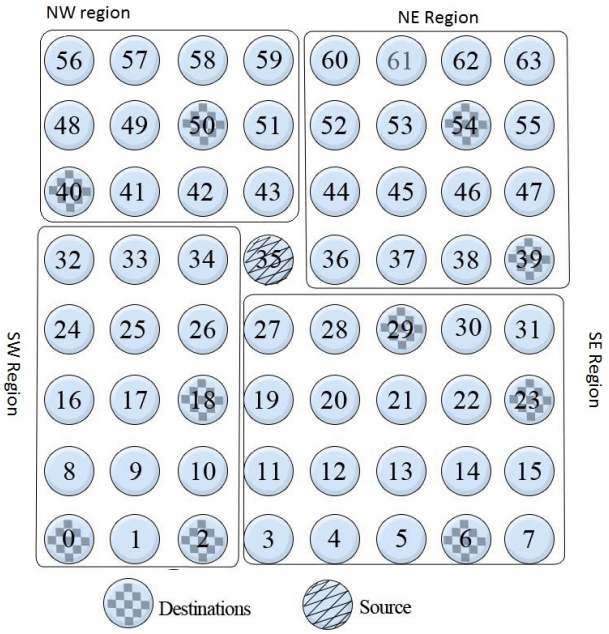

Figure. 3: Destination list and region based mask based on a source.

can be represented using n-bit string method where each bit corresponds to node number in an 'n' core system. For a 8x8 network, the multicast field has a 64-bit field. A 32-bit physical address field which indicates the address of memory location (assuming 4 GB physical address space).

Figure 2 depicts the header information of multicasting scenario mentioned in Figure 3. With respect to the source node 35, the network is partitioned into four different regions (NE, SE, NW, SW) based on the condition specified in the beginning of this section. Corresponding region mask is obtained by setting the bit positions of the routers that belong to the specific sub-regions. In a NW_{Region_mask} bit positions are set for those routers that falls in NW_{Region}. It is assumed that there are 10 destinations in the network, denoted as $D_{List} = \{0, 2, 6, 18, 23, 29, 39, 40, 50, 54\}$. For each of these values, the corresponding bit position is set as 1 in the multicast destination header list.

V. PROPOSED ARCHITECTURE

The 2-cycle router pipeline framework of the proposed architecture is given in Figure 4. In the first stage ejection and duplication unit, prioritization and preemption unit and routing and duplication unit are present. The second stage consists of the port allocation unit. Features of the functional units are explained in detail in the following section. The four internal flit channels (North, East, South, West) carry the incoming flits arriving from the neighbouring routers and move them through the router's pipeline units. The pipeline registers(A, B and C) act as a latch and are used for the smooth transition between the stages. The pipeline register B contains not only the flits from the neighbouring router but also their duplicates.

The proposed architecture implements hardware based multicast functionality on router without adding significant overhead on the router pipeline. Both the multicast and the unicast messages use the same router architecture for routing the flits. For efficient routing of multicast flits, routers need the duplication component in it. The partitioning mechanism is employed by the router's duplication logic so that the message reaches the intended multicast destinations in the header list. There are two functional units that employ duplication logic; one in the ejection unit and the other in the routing and duplication unit.

The router architecture use minimal pool of buffer called side buffer for accommodating the preempted flits thus making a free channel for the flit duplication. The core buffer contains the flits injected by the local core.

A. Ejection and Duplication Unit (EDU)

Ejection takes place only if the flit reaches its destination. If the current router address and the destination address (in the packet header) show the same value, ejection of the flits to the local processing core happens. In case of a unicast message, the flit is removed from the internal flit channel for making a free channel and then gets ejected through the ejection port.

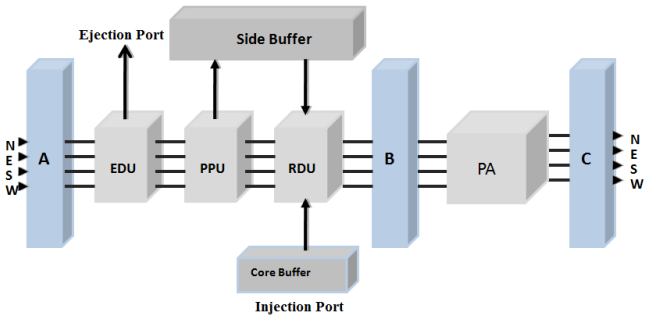

Figure. 4: Deflection Router Architecture that supports multicasting.

If the muticast flit contains more than one destinations then the processing is different. In such scenario, if the multicast message reaches its destination or any of its destination, then it is serviced by providing a copy of that packet. At the destination router, a duplicate of the flit is delivered to the core and the packet is forwarded to the next destination. The original flit is forwarded to the next destination in the destination list. The multicast destination header list of original message is updated to avoid the redundant flits to the routers. The original flit still continues its advancement through the network, except the one that got serviced in the destination header list.

B. Prioritization and Preemption Unit (PPU)

In order to ensure fair progress of both multicast as well as regular flits in a network, the prioritization of all the incoming flits based on a priority matrix is performed. Highest priority is assigned to a multicast packet and the remaining flits are sorted based on the hops to the destination.

There can be multiple multicast flits in a router and all may not get a provision for duplication in the current router. If a flit wants to be duplicated at this router, a free channel is made possible by a random pick (from the unicast message). One flit from the internal flit channels in the pipeline is forcefully preempted to the side buffer. By this preemption, a free channel is created thereby making a provision for the duplicate flit. The starvation of the flits residing in the side buffer and core buffer are avoided by re-injecting them back into the router pipeline during the subsequent cycles.

C. Routing and Duplication Unit (RDU)

The Duplication and Routing Unit consist of two sub-modules; Duplication unit followed by the Routing unit. In case of multicast message, the destinations mentioned in the header list can be effectively reached, only if we create duplicate flit at the current router. The duplication of the original message with a new set of destinations specific to the regions is done.

The duplication logic is a light weight control module that consumes very less area. As soon as the flits reach the duplication logic, the multi-destination list is extracted. To implement the duplication logic, an enable signal for each of the pipeline channel is maintained. Initially enable signal is reset to zero for all the pipeline channels. If any one of the flits is a potential candidate for duplication, then the corresponding duplication logic is activated by setting the enable signal. The potential candidate for the duplication is found based on the congestion metric obtained from the *Internal Channel* and *Flit Count* which will be explained in the following section.

The enable signal is set for a flit depending on the congestion metric value. Congestion metric takes into account the free *Internal Channel* and *flit count*. Duplication of the flits at the source router or in the intermediate destinations is done only after considering congestion value.

Internal Channel

Flits flow through the internal flit channel in a router. Since there are four input ports, maximum flits that can move through the router's internal flit channel is restricted to four. A replication happens only if there is a vacant channel for the flit to reside in it. The difference between the available output ports and the required ports for the incoming flits gives the total number of free internal flit channel of a router.

If there are vacant channels and if more than one flits are designated for duplication, the flit appropriate for duplication is resolved after analyzing the *Flit Count*. This is done to avoid the congestion overhead due to duplication.

Flit Count

Flit count is the total number of flits that have been passed to the specific direction ($N,\ E,\ S\ and\ W$) through that router. Flit count denotes the history based scheme that captures the number of flits to collect the congestion information. If the number of flits that get propagated to a unique direction is prominent, duplication will add extra traffic which may lead to contention.

A set of four (one for each direction) 5-bit counters are employed to keep track of the flits for each direction. The counter value is refreshed to 0 after a C number of clock cycles. The counter keeps track of the flits owing through the router during this C cycle interval to set a 2-bit priority value. Four different priority levels are set for each output ports: Level 00 for counter values less than 10; level 01 for counter values between 11 and 15; level 10 for counter values between 16 and 20 and level 11 for counter values greater than 21. A fair congestion information at a router helps in regulating the duplication at a router.

The direction with the highest priority (Level 00) is considered to be less congested and the flit with which requests for these direction are given the provision for duplication. This information is passed on to the duplication unit so as to enable the signal. Now the duplication unit generates the duplicate for the each possible partitions.

The duty of the duplication logic is not only restricted

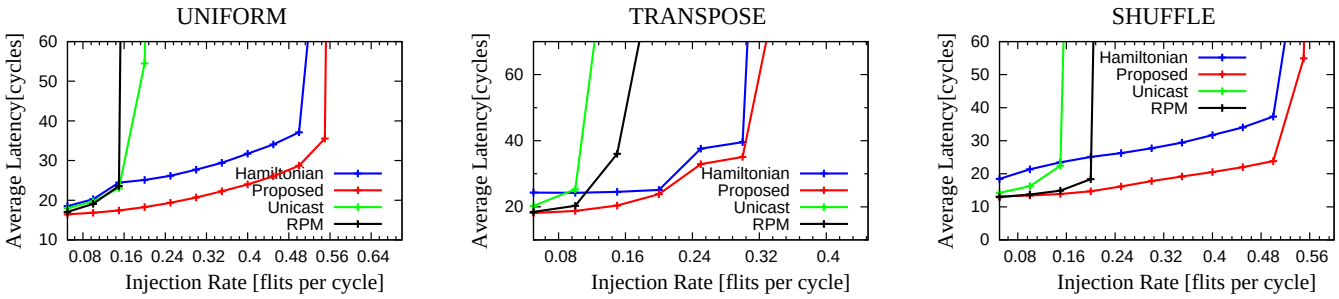

Figure. 5: Comparative analysis of latency versus injection rate for various synthetic traffic patterns in 8x8 mesh network.

to duplicate a packet, but also to modify the original packet header. This prevent redundant flits from reaching the router. Thus the packet header of the new packet has only the disjoint subset of the destination header nodes. The logical AND operation between the $region\_mask$ and the *destination header list* results in setting the bit positions exclusively for a particular partition.

To avoid the storage overhead, the duplicate flits are placed into the pipeline register which get forwarded to the free pipeline flit channel. There is not much area overhead for the current router due to the duplication scheme. Actually, the replication component in our router is only a control logic (for each internal channel). Hence it will not consume considerable area.

After obtaining the new multicast destination header list, the packet is forwarded to the new target in the destination list. The destination address is fixed as the nearest router among the new multicast destination header list.

The routing unit compute the next feasible output port to forward the flits in the current router. Route computation logic in the routing unit employs XY routing to find the next router to forward the flits. Core buffer holds the preempted flits from the previous cycle. To avoid the starvation of the flits residing in the core and the side buffer they are allowed to enter into the router pipeline after a threshold time of three-cycle.

D. Port Allocation

Port Allocation Unit consists of two 2 x 2 arbiters similar to one used in CHIPPER [12]. Regular flits together with the duplicated flits (maximum of four) from the pipeline register moves to the port allocation stage. It allocates the output ports to each of the incoming flits based on the information obtained from Routing and Prioritization Unit. The highest priority flit always gets the desired output port. Other flits may get deflected or assigned productive ports based on port conflicts in each stage of the arbiter.

VI. EXPERIMENTAL METHODOLOGY

We use NoC simulator, Booksim [15], to model the two-stage multicast deflection router architecture by modifying the traditional input buffered simulator. We employ single flit scheme such that every flit in the network contains all the information required for routing since all the flits get routed independently. In order to show the performance improvement of the proposed work, we compare its performance with the state-of-the-art existing multicasting techniques. In the first technique, multicasting is implemented as independent unicast packets. The second technique employs a Hamiltonian path based multicasting, which performs a destination list sorting at the source router. This method service all the destinations by making a duplicate only at the destination router. Next is the RPM [7] where we recursively make partitions for each regions.

We consider synthetic workloads for evaluation of our proposed router design. We evaluate our design using several standard synthetic traffic patterns: uniform, shuffle and transpose for 8x8 mesh network under 4% multicast traffic. Average flit latency, average transaction latency of multicast packets and average link traversal count values are collected for each synthetic traffic pattern, with varying injection rate, until the saturation point is reached.

VII. EXPERIMENTAL ANALYSIS

A. Effect on Average Flit Latency

The latency of a flit is defined as the number of clock cycles required to traverse the network from the source to destination. We analyse the results under 4% multicast traffic in such a way that, at every 25 cycles one multicast message is generated with 8 destinations. Figure 5 shows the plot of injection rate vs average flit latency for various synthetic traffic patterns in 8x8 mesh network under 4% multicast traffic. The average latency takes into account the effect of both multicast and unicast messages.

Average latency increases with increase in the injection rate. A point in the injection rate at which the latency increases exponentially for linear increase in injection rate is called the saturation point. Port contention increases with increase in the injection rate, which results in the increased latency. For all the synthetic traffic patterns, the proposed work saturates at high injection rate than the Hamiltonian [20] and Multicast as unicast method. The RPM and the proposed work shows almost same latency at very low injection rate, but RPM saturates at an early injection rate due to the flooding of duplication for each regions. From this

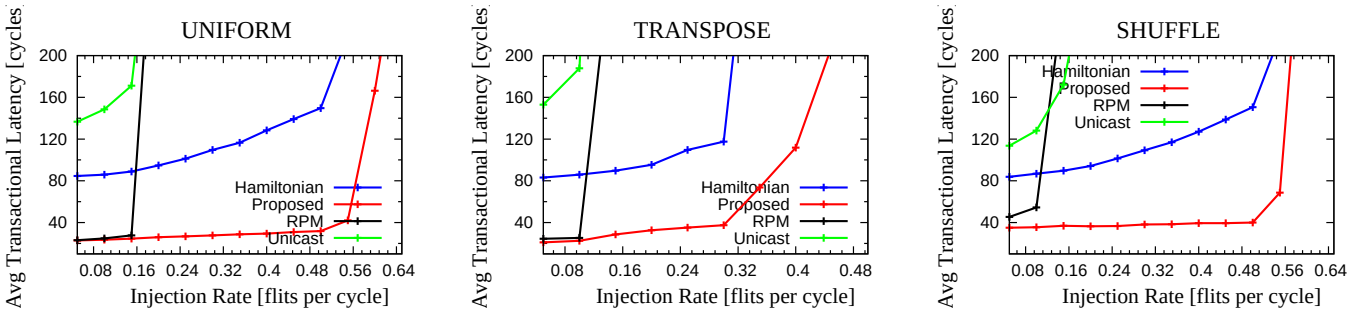

Figure. 6: Comparative analysis of Average Transaction Latency versus injection rate for various synthetic traffic patterns in 8x8 mesh network.

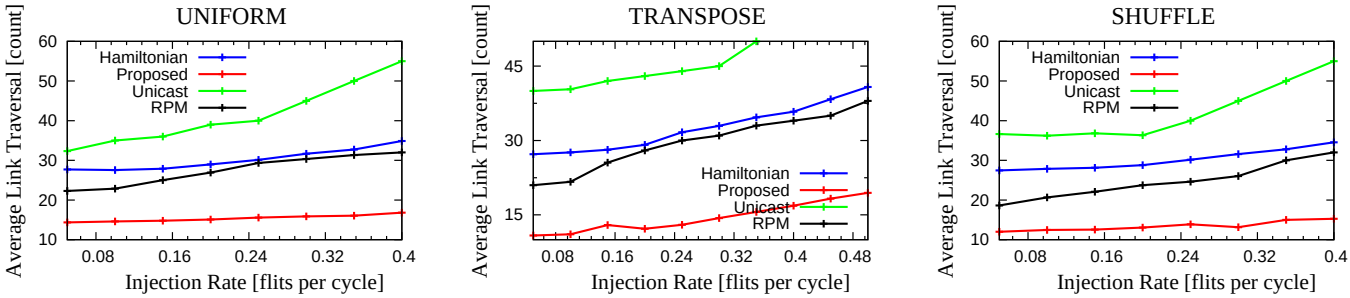

Figure. 7: Comparative analysis of Average Link Traversal versus injection rate for various synthetic traffic patterns in 8x8 mesh network.

observation, we can conclude that our proposed technique is capable of working at high injection rate.

B. Effect on Average Transaction Latency

Multicast transaction latency is defined as the time cycles needed by a source router to disseminate the multicast message to all its multicast destination routers. In other way, the transaction latency can be considered as the time between the creation of the multicast message to the dispatch of this multicast message to the last destination. For multi-threaded applications, a low transaction latency is a significant requirement for the application in the source to continue its execution. Another significant requirement for low transaction latency is that, the multithreaded applications demand for fast propagation of the invalidation message to be propagated to its sharer destinations.

Figure 6 shows the plot of injection rate vs average transaction latency for various synthetic traffic patterns in 8x8 mesh network under 4% multicast traffic. We can see that the proposed method attains significant reduction in transaction latency with respect to other technique right from the zero load to saturation.

There is a wide difference in the transaction latency of the proposed with all other techniques. This is because, in the proposed scheme we assign high priority to the muticast packet which helps in the productive port allocation of these packets. Since it is assigned productive port which results in the timely and effective delivery of the multicast message at all its destination. At each router the unicast flits are penalized for the multicast flit by moving to the side buffer.

C. Effect on Average Link Traversal Count

For a multicast message, the link traversal count is defined as the total count of link traversed by the message through the network before reaching all its destinations. The link traversal count depends on - 1) number of duplicates created for the message under consideration 2) hops covered per message 3) Injection rate.

A flit encounters deflections in many intermediate routers before it reaches the destination. As the injection rate increases, the number of port conflict at a router also increases. So an increase in the contention increases the deflection rate which in turn increases the link traversal count. The number of times a flit gets deflected affects its hops. A packet that has encountered a deflection needs minimum of three hops to come back to its original path. Figure 7 shows the plot of injection rate vs average link traversal count for various synthetic traffic patterns in 8x8 mesh network under 4% multicast traffic.

Results indicate that our proposed technique reduces the number of hops for the packet delivery, which will pave way for the reduction of dynamic power emission in an NoC system. When the network load increases the deflections occurs more frequently which leads to performance degradation. For the proposed method the improved priority scheme reduces the deflection rate. Thus the multicast flits with less deflection rate play an important role in reducing the link traversal count.

Table 1: Routing pipeline delay analysis of proposed work using verilog synthesis.

UNIT	LATENCY
EDU	0.63ns
PPU	0.45ns
RDU	0.8ns
PA	2.022ns

D. Effect on Router Pipeline Latency

Verilog model of the proposed system is synthesized using Xilinx ISE 14.1. This gives the router latency for each unit in the pipeline stage. The Table 1 shows the propagation latency of each unit for the proposed architecture generated from the synthesis results given by Synopsys Design Compiler with 90nm cmos technology [24]. We have found that our proposed model achieve 12.5 % reduction in latency with respect to RPM and 9 % with respect Hamiltonian based.

VIII. CONCLUSION

Through this work, an energy efficient router framework for buffer-less multicasting has been recommended. The proposed architecture is compared with Hamiltonian based, Multicast as Unicast and Recursive Partitioning methods. The recommended architecture is much faster in delivering multicast messages to all its destinations. Experimental results across all the synthetic traffic pattern show that, the proposed approach is efficient enough to be chosen as the best choice for multicasting in buffer-less NoCs. New design has the least Average Flit transaction latency, Average latency and also the Average Hop Count when compared with all other techniques. Thus the proposed architecture is an efficient design choice for buffer-less NoCs with multicast messages even at high traffic.

References

[1] C. Chiang and L. Ni, Multi-address encoding for multicast, *In Proceedings of the Workshop on Parallel Computer Routing and Communication Workshop*, pp. 146-170, May 1994.

[2] M. P. Malumbres, J. Duato, and J. Torrellas, An efficient implementation of tree-based multicast routing for distributed shared-memory multiprocessors, *In Proceedings of the 8th IEEE Symposium on Parallel and Distributed Processing*, pp. 186-190, October 1996.

[3] N.E. Jerger, L.S. Peh, M. Lipasti, Virtual circuit tree multicasting: a case for on- chip hardware multicast support, *Proceedings of the 35th International Symposium on Computer Architecture*, pp. 229-240, 2008.

[4] I.V. Senin, L. Mhamdi, K. Goossens, Efficient multicast support in buffered crossbars using networks on chip, *in Proceedings of the 28th IEEE conference on Global telecom- munications*,pp. 5024-5030, 2009.

[5] P. Abad, V. Puente, J.A. Gregorio, MRR: enabling fully adaptive multicast routing for CMP interconnection networks, *In the Proceedings of the 15th International Conference on High-Performance Computer Architecture*, pp. 355-366, 2009.

[6] F.A. Samman, T. Hollstein, M. Glesner, Planar adaptive router microarchitecture for tree-based multicast network-on-chip, *In the of the International Workshop on Network on Chip Architectures*, vol. 36, no. 6, pp. 449-461, 2012.

[7] L. Wang, Y. Jin, H. Kim, E.J. Kim, Recursive partitioning multicast: a bandwidth-efficient routing for on-chip *In the Proceedings of 3rd ACM/IEEE International Symposium on Networks-on-Chip*, pp.64-73, 2009.

[8] Bahrebar P, Stroobandt D, Improving Hamiltonian-based routing methods for on-chip networks: a turn model approach, *In the Proceedings ACM/IEEE on Design Automation and Test in Europe, DATE*, pp. 1-4, 2014.

[9] Chwiebert.L, Jayasimha D.N, A necessary and sufficient condition for deadlock-free wormhole routing, *Journal of Parallel and Distribution Computing*, 32, 1, pp.103-117, 1996 .

[10] Fleury E, Fraigniaud P, A general theory for deadlock avoidance in wormhole-routed networks, *IEEE Transaction on Parallel Distributed System* 9, 7, pp.626-638, 1998.

[11] C. M. Chiang and L. M. Ni, Deadlock free Multi-Head Wormhole Routing, *In Proceedings of The First High performance Computing-Asia*, 1995

[12] C. Fallin, C. Craik, O. Mutlu. Chipper: a low-complexity bufferless deflection router *in High Performance Computer Architecture (HPCA), IEEE 17th International Symposium on, IEEE*, pp. 144-155, 2011.

[13] C. Fallin, G. Nazario, X. Yu, K. Chang, R. Ausavarungnirun, O. Mutlu. Minbd: minimally-buffered deflection routing for energy-efficient interconnect, in: Networks on Chip (NoCS) *Sixth IEEE/ACM International Symposium on, IEEE*, pp. 1-10, 2012.

[14] J. Jose, B. Nayak, K. Kumar, M. Mutyam. Debar: deflection based adaptive router with minimal buffering, *in: Design, Automation and Test in Europe Conference and Exhibition (DATE) IEEE*, pp. 1583-1588, 2013.

[15] N. Jiang 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, pp. 86-96, 2013.

[16] M. Tomasevic and V. Milutinovic, The Cache-Coherence Problem in Shared-Memory Multiprocessors: Hardware Solutions, 1994.

[17] W. Zuo, S. Feng, Z. Qi, J. Weixing, L. Jiaxin, D. Ning, X. Licheng, T. Yuan, Q. Baojun, Group-caching for NoC based multicore cache coherent systems*in Proceedings of the Conference on Design, Automation and Test in Europe, DATE09*, pp. 755-760, 2009.

[18] K. Strauss, X. Shen, J. Torrellas, Uncorq: unconstrained snoop request delivery in embedded-ring multiprocessors, *in Proceedings of the 40th Annual IEEE/ACM International Symposium on Microarchitecture MICRO07*, pp. 327-342, 2007.

[19] M.R Arun, P.A Jisha, J. Jose, Novel Energy Efficient Multicasting Approach For Mesh NoCs *in Procedia Computer Science 93*, pp. 283-291, 2016

[20] P Bahrebar, D Stroobandt, Hamiltonian-based odd-even turn model for maximally adaptive routing in 2D mesh networks-on-chip, *Computers and Electrical Engineering*, pp. 386-401, 2015

[21] J. Duato, A new theory of deadlock-free adaptive routing in wormhole networks, *IEEE Transactions on Parallel and Distributed Systems* vol. 4, no. 12, pp. 1320-1331, December 1993

[22] 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.

[23] T. Moscibroda and O. Mutlu, A case for bufferless routing in on-chip networks,*in Proceedings of the Annual International Symposium on Computer Architecture (ISCA)*, pp. 196-207, 2009

[24] www.synopsys.com

Author Biographies

Elizabeth Isaac is an Assistant Professor at M A College of Engineering Kothamangalam, Kochi, Kerala. She is a post-graduate in Computer Science and Engineering from VIT University. She did her graduation in Computer Science and Engineering from M A College of Engineering Kothamangalam. She is currently a Ph.D. Scholar in the School of Computing Science and Engineering, VIT University, Vellore

M.Rajasekhara Babu is a Senior faculty member at School of Computing Sciences, VIT University, Vellore, India. He completed his Ph.D from VIT University. He receives his Bachelors in Electronics and Communication Engineering from Sri Venkateswara University, Tirupathi, India and took his Masters in Computer Science and Engineering from Regional Engineering College (NIT), Calicut. His areas of interest include multi core architectures, compilers

John Jose is an Assistant Professor at Department of Computer Science and Engineering, Indian Institute of Technology Guwahati, Guwahati, India. He did his B.Tech from Cochin University and M.Tech from VIT University, Vellore He completed his Ph.D from Indian Institute of Technology, Madras. His areas of interest include computer architecture, interconnection networks and high performance computing.