An Efficent Dynamic Multicast Routing Protocol for Distributing Traffic in NOCs

Masoumeh Ebrahimi[†*], Masoud Daneshtalab[†*], Mohammad Hossein Neishaburi[†], Siamak Mohammadi[†],
Ali Afzali-Kusha[†], Juha Plosila[*], Hannu Tenhunen[*]
[†]*Nanoelectronics Center of Excellence, School of Electrical and Computer Engineering*
University of Tehran, Tehran, Iran
[*]*Department of Information Technology, University of Turku, Turku, Finland*

{masebr, masdan, juha.plosila, hannu.tenhunen}@utu.fi, {mh.nisha, smohamadi, afzali}@ut.ac.ir

Abstract- *Nowadays, in MPSoCs and NoCs, multicast protocol is significantly used for many parallel applications such as cache coherency in distributed shared-memory architectures, clock synchronization, replication, or barrier synchronization. Among several multicast schemes proposed in on chip interconnection networks, path-based multicast scheme has been proven to be more efficient than the tree-based, and unicast-based. In this paper a low distance path-based multicast scheme is proposed. The proposed method takes advantage of the network partitioning, and utilizing of an efficient destination ordering algorithm. The results in performance, and power consumption show that the proposed method outstands the previous on chip path-based multicasting algorithms.*

1. Introduction

Since the traditional bus-based communication solutions are not useful anymore, new communication architecture is needed. Network on Chip (NoC) has been addressed as a solution for the communication requirement and difficulty of global interconnections [1]. However, on-chip networks do not need to follow the all standard schemes for communication in computer networks since they can use lighter and faster protocol layers. Therefore, the Communication in NoC (or MPSoC) can be either unicast (one-to-one) or multicast (one-to-many) [2]. In unicast a message is sent from a source node to a single destination node, while in multicasting a message is sent from a source node to an arbitrary set of destination nodes. Many MPSoC applications such as replication [3], barrier synchronization [4], cache coherency in distributed shared-memory architectures [5], and clock synchronization [6] employ multicast protocol. Although Multicast communication can be implemented by multiple unicast communications, this alternative method degrades the performance and increases the congestion in the network [7].

In this paper, we propose an efficient path-based multicast wormhole routing scheme in 2D-mesh on chip network [7] which takes advantage of the odd-even turn model [10] and congestion detection. The odd-even turn model is an adaptive and deadlock-free unicast wormhole routing algorithm in NoC [9][11]; therefore it is deadlock-free and does not require virtual channels [9][10] [11]. In fact, adding virtual channel is costly since the complexity and latency of the controller increase with the number of virtual channels due to increased buffering and arbitration requirements. Besides, in the proposed multicast algorithm the destination addresses are sorted in a low distance order by an efficient ordering algorithm, and then placing this sorted list in the header flit(s) of the message. Afterwards the message routes towards the destinations such as unicast adaptive wormhole routing which is based on the odd-even turn model. Experimental results with multicast traffic profile show that power and performance can be improved significantly by using the proposed multicast algorithm compared to the traditional path-based multicast algorithm such as Dual-path, Multi-Path, and Column path. Furthermore, the chip area overhead is marginal and at the greatest is %8. The paper is organized as follows. In Section 02, a brief review of the related works is presented. In Section 03, the proposed and traditional path-based multicast algorithms are discussed while the proposed switch architecture is presented in section 4. The results are discussed in Section 5 with the summary and conclusion given in the last section.

2. Related works

Multicast routing algorithm can be classified as unicast-based [13], tree-based [13], and path-based [7]. In unicast-based, the multicast operation is performed by sending a separate copy of message from the source to every destination or, alternatively, by sending the unicast message to subset of destinations. The drawback of this scheme is on account of the fact that multiple copies of the same message are injected into the network, the traffic of the network will be increased. Furthermore, each copy of message loses considerable startup latency at the source.

The tree-based multicast approach, a spanning tree constructed in which the source is indicated as the root and then messages are sent down the tree. In this way a message might be replicated at some of the intermediate nodes and forward along multiple outgoing channels toward disjoint subsets of destinations. If one branch of the tree is blocked, all are blocked. Branches must proceed forward in lock step, which may cause a message to hold many channels for extended periods, resulting in increasing network contention. Although such schemes have to be used effectively in networks employing store-and-forward and virtual cut-through routing, tree-based routing incurs high congestion in wormhole networks [7]. A solution to overcome the tree-based disadvantage is to utilize the path based multicast wormhole routing. In this method, a source node prepares a message for delivery to a set of destinations by first sorting the addresses of the destination in order in which they are to be delivered, and then placing this sorted list in the header of the message. When the header entered a router with address A, the router checked to see if A is the next address in the header. If so, the address A is removed from the message header and a copy of date flits will be delivered to the local core and the flits are forwarded to the next node on the path. Otherwise, the message is forwarded only to the next node on the path. In this way, the message is eventually delivered to every destination in the header. A number of studies have shown that a path-based exhibit superior performance characteristic over their unicast-based and tree-based counterparts [14]. In this work, routers use the adaptive odd-even turn model for routing the message among destinations, which leads the path-based to be adaptive and deadlock free. None of the path-based multicast techniques has addressed the issue of using an optimized sorting algorithm into a new multicast scheme. By applying the proposed sorting algorithm we can achieve better power performance compared to the traditional path-based multicast algorithm with low hardware overhead.

3. Path-based Multicast Routing

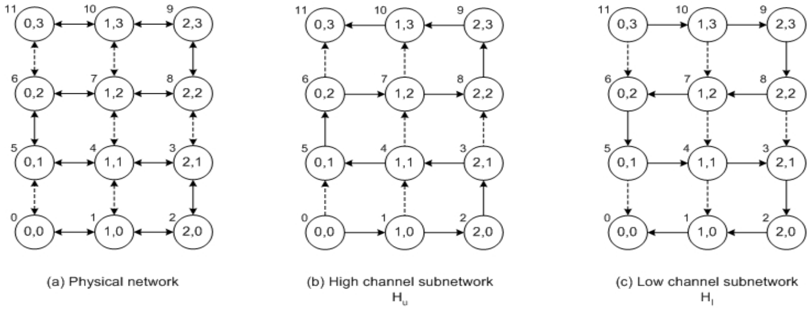
In this paper, we consider two-dimension meshes with wormhole switching technique. Networks with The two-dimensional (2D) mesh topology offer massive parallelism and are more scalable than many other approaches to MPSoC interconnection [8]. Besides, Meshes are suited to a variety of applications including matrix computation, image processing and problems whose task graphs can be embedded naturally into the topology [15]. Formally, an $m{\times}n$ 2D mesh consists of $N= m{\times}n$ nodes; each node has an associated integer coordinate pair (x,y), $0{\leq}x{<}n$ and $0{\leq}y{<}m$. Two nodes with coordinates (xi,yi) and (xj,yj) are connected by a communication channel if and only if $|xi - xj| + |yi - yj| =1$.

The path-based routing algorithm is established as the Hamiltonian path algorithm [7]. In this method an undirected Hamiltonian path of the network is constructed. A Hamilton path visits every node in a graph exactly once. In this algorithm, for each node in an m×n mesh a label $L(x, y)$ is assigned where x and y are node's coordinates, as follows: $L(x, y) = y \times n + x$, if y is even, and $L(x, y) = y \times n + n - x - 1$, if y is odd. As exhibited in Fig. 1, two directed Hamilton paths (or two subnetworks) are constructed by the labeling [7]. The high channel subnetwork (H_u) starts at (0, 0), and the low channel subnetwork (H_l) ends at (0, 0). In case the label of the destination node is greater than the label of the source node, the routing always takes place in the H_u subnetwork; otherwise it takes place in the H_l subnetwork. The destinations are placed into two groups. As exhibited in Fig. 1, two directed Hamilton paths (or two subnetworks) are constructed by the labeling [7]. The high channel subnetwork (H_u) starts at (0, 0) ,and the low channel subnetwork (H_l) ends at (0, 0). In case the label of the destination node is greater than the label of the source node, the routing always takes place in the H_u subnetwork; otherwise it takes place in the H_l subnetwork. The destinations are placed into two groups. One group contains all the destinations that could be reached using the H_u subnetwork, and the other contains the remaining destinations, that could be reached using the H_l subnetwork. To reduce the path length the vertical channels that are not part of the Hamilton path (the dashed lines in the Fig. 1) could be used in appropriate directions. The proposed Dual-Path (DP) [7], Multi-Path (MP) [7] and Column-Path(CP) [14] algorithms that have been described in the following, use the Hamiltonian path strategy.

DP and MP Multicast Routing: In Dual-Path (DP) routing algorithm the destination node set is partitioned into two subsets, D_H and D_L, where Every node in D_H has a higher label than that of the source node and every node in D_L has a lower label that that of the source node. D_H and D_L which are then sorted in ascending order and descending order with the label of each node used as the key for sorting, respectively. Thus, multicast messages from source node will be sent to the destination nodes in D_H using the H_u subnetwork and to the destination nodes in D_L using H_l subnetwork. Consider the example illustrated in Fig. 2(a) for a 6×6 mesh network where node (2, 3) will be sent its multicast messages to destinations (2, 0), (4, 0), (0, 1), (2, 1), (4, 1), (0, 4), (5, 4), (3, 5) and (5, 5). Accordingly, two subsets are organized. The first subset (D_H) that has all the destinations that could be reached from the source node using H_u subnetwork which are (0, 4), (5, 4), (3, 5) and (5, 5) in sequence and the second one (D_L) has the remaining destinations that could be reached using the H_l subnetwork which are (2, 0), (4, 0), (4, 1), (2, 1) and (0, 1). Some of the Vertical links that are not part of the Hamiltonian paths are used properly, for minimizing the paths. In order to reduce the path lengths Multi-Path (MP) multicast routing algorithm has been proposed. In this scheme as most nodes have four output channels in 2D mesh, up to four independent paths can be used to deliver a message, in which the destination sets D_H and D_L of the dual-path are also partitioned. The set D_H is divided into two subsets, one consist of the nodes whose x coordinates are greater than or equal to that of the source and the other subsets containing the remaining nodes in D_H. The set D_L is partitioned in a similar way. Hence, all the destinations of the multicast message are grouped into four disjoint subsets such that all the destinations in a subset are in one of the four quadrants when source is taken as the origin. For the multi-path example shown in Fig. 2(b), the destination set is first divided into two sets D_H and D_L at source (2, 3) , as D_H={(0, 4),(5, 4),(3, 5),(5, 5)} and D_L={(2, 0),(4, 0),(4, 1),(2, 1), (0, 1)}. As exhibited in Fig. 2(b), D_H is divided into two subsets, which are D_{H1}= {(0, 4)} and D_{H2}= {(5, 4), (3, 5), (5, 5)}. In the same way D_L is divided into two subsets, with D_{L1}= {(0, 1), (2, 1), (2, 0)} and D_{L2}= {(4, 0), (4, 1)}. The dual-path and multi-path are both deadlock free and could be used for unicast and multicast routing simultaneously [7].

CP Multicast Routing: In this method, the destination node set is partitioned to 2k subsets; k is the number of columns in the mesh, and at most two messages will be copied to each column. If a column of the mesh has one or more destinations in rows above that of the source then one copy of the message is sent to service all those destinations. Similarly, if a column has one or more destinations in the rows below that of the source, then the another copy of the message is sent to service all those destinations. One copy of the message is sent to a column if all destinations in that column are either below or above the source node; otherwise, two messages are sent to that column. For instance, to send a message to destinations (2, 0), (4, 0), (0, 1), (2, 1), (4, 1), (0, 4), (5, 4), (3, 5) and (5, 5) from source node (2, 3) using the Column-Path (CP) routing algorithm based on Hamiltonian-path-based is shown in Fig. 2(c). Six copies of the message are used to achieve the desired multicast operation. Though destinations (0, 1) and (0, 4) are in the same column, two message copies are sent to this column, since two of the destinations are above the source node's row and the other below. The routing algorithm is used by this scheme is based on the row-column routing algorithm. Therefore, the CP routing algorithm is compatible with the unicast routing method and it is deadlock-free and livelock-free. The column-path multicast algorithm uses short paths and performs better in high-loads circumstances. However, it suffers from high network latencies due to its excessive number of start-ups.

Proposed Path-based Multicast Routing: Tree aspects are considered by this scheme. First, it is utilized of network partitioning similar to multi-path multicast routing in which up to four groups could be formed. Second, the ordering of the destinations in the path must be optimized for a shortest multicast path. For this propose, the algorithm shown in Fig. 3 is proposed. In this algorithm, for each node a label is assigned a label as $L(x, y) = y \times n + x$. Similar to the Multi-Path multicast algorithm destination node set is partitioned into four subsets, D_{H1}, D_{H2}, D_{L1} and D_{L2} which are then sorted in low-distance order with the distance vector of each node used as its key for sorting (distance vector of each node is computed as $k = |y-y_0|+|x-x_0|$). For low-distance order algorithm in the origin set, the node (v) which has the lowest distance vector as compared to the source node (u_0) is placed in the Temp_set and removed from the origin set; then the selected node will be assigned as the source node. While the origin set is not empty, this sequence must be repeated otherwise; otherwise the Temp_set is placed in the origin set. If there are two nodes with an equal distance vector compare to the source node, the one with lesser x dimension compared to that of the source node would be selected first. Subsequently, the origin set will be laid in the message header.


(a) Physical network

(b) High channel subnetwork H_u

(c) Low channel subnetwork H_l

Fig. 1. (a) A 3×4 mesh physical network with the label assignment and the corresponding (b) high channel and (c) low channel networks. The solid lines indicate the Hamiltonian path and dashed lines indicate the links that could be used to reduce path length in routing.

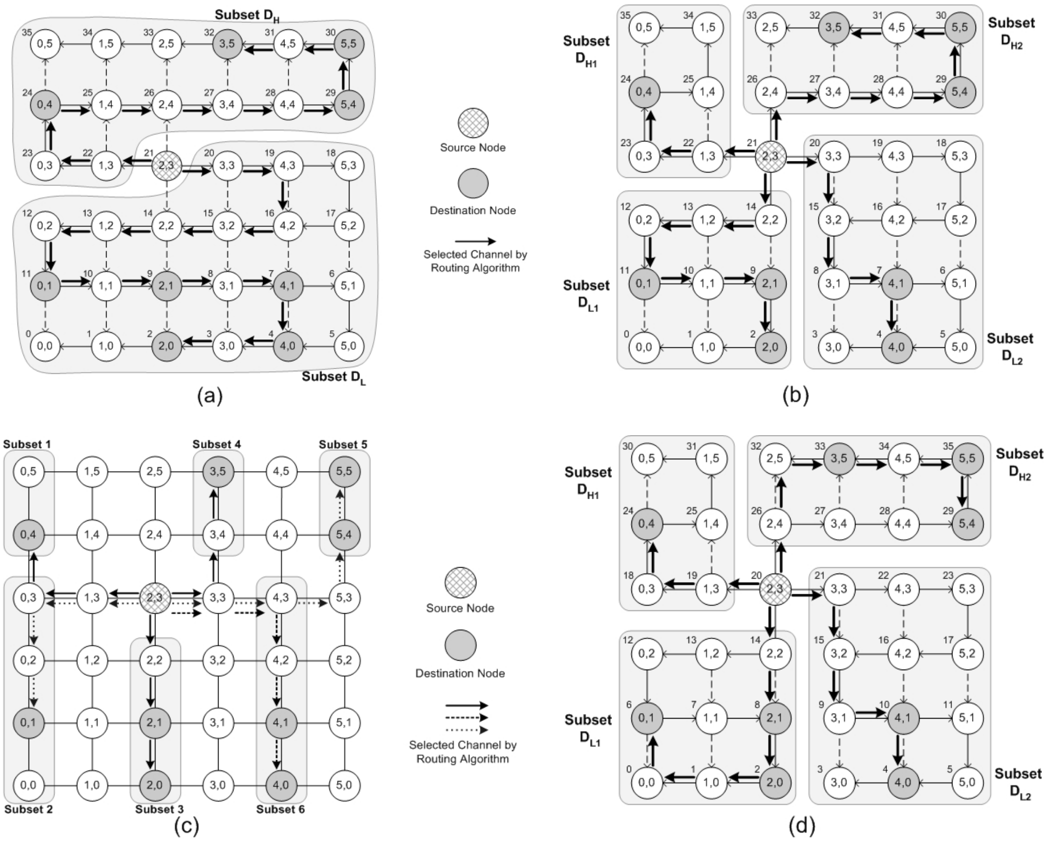


Fig. 2. Examples of (a) Dual-path (DP) [7], (b) Multi-path (MP) [8], (c) Column-Path (CP) [14], and (d) proposed multicast routing from (2, 3). The unused links are not indicated.

Third, in the proposed method the routing algorithm used to route the multicast message is based on the odd-even turn model adaptive routing algorithm [11]. Therefore, both unicast and multicast messages route adaptively through the destination(s) .The following is a case in point: consider the example used for MP, DP, and CP. The paths used under this proposed algorithm are shown in Fig. 2(d). Assume that in Fig. 2(a) a multicast message from node (1, 3) will be sent to destinations (0, 3), and (0, 2). After delivering the message to the destination node (0, 3) the message should be sent through the next destination node (0, 2) which causes a forbidden turn [11] in the odd-even model. Under this circumstance, a copy of the message will be sent to the local input buffer by the core, for keeping the network deadlock free, while the message delivers to the local core. In fact, the message will be sent to the next destination node (0, 2) by the previous destination core (0, 3).This mechanism is called absorb-and-retransmit which would increase the latency. But since we have measured the number of forbidden turns occurring in the network under different traffic patterns, different network sizes, and various multicast destinations, the average number of forbidden turns is at most 7% of all turns in the network. Therefore, as shown in the experimental result section, these forbidden turns, causing the absorb-and-retransmit mechanism have not considerably degraded the performance of the network.

4. Hardware Implementation

In this work, due to scalability, cross-section bandwidth, and the fixed degree of nodes, we make use of an n × n network of interconnected tiles with a mesh topology[8][15]. Each tile is composed of a PE (Processing Element) and a router connected to its four adjacent routers in addition to the PE of the tile through some channels. Two unidirectional point-to-point links form the channel. To minimize the delay and the required resources, we have used wormhole method for the switching. In this method, a message is divided into smaller segments called FLITs (FLow control digit) which are routed successively until they reach their destination [8].

Message Format: The multicast message format is shown in Fig. 4. As it can be seen; it includes a header flit and a parametric number of payload flits. Each flit is *n* bit wide and the nth bit is the EOM (End Of Message) sign and the *(n-1)th* bit is the BOM (Begin Of Message) sign.

In the header, the third field is used to describe the type of the message. There are two types of message: unicast (T=0) and multicast (T=1), represented by T. The specific addresses of the source node and the destination node(s) are placed in the last field of the header, respectively and the content of the message is located in the rest of flits (Payload).

```
Algorithm: Ordering and partitioning the destination set
Inputs: Destination set D;  source node ($(x_0, y_0)$);
        distance table T;
Outputs: Sorted destination sets $D_{H1}$ ,$D_{H2}$ , $D_{L1}$, $D_{L2}$
         for 4 multicast paths.
Begin
1. For every node assign a label as: $L(x,y) = y \times n + x$
2. $D \rightarrow \{D_H, D_L\}$;  $D_H = \{(x,y) \mid L(x,y) > L(x_0,y_0)\}$;
                                  $D_L = \{(x,y) \mid L(x,y) < L(x_0,y_0)\}$;
3. $D_H \rightarrow \{D_{H1}, D_{H2}\}$;  $D_{H1} = \{(x,y) \mid x < x_0, y \geq y_0\}$;
                                          $D_{H2} = \{(x,y) \mid x \geq x_0, y > y_0\}$;
   $D_L \rightarrow \{D_{L1}, D_{L2}\}$;  $D_{L1} = \{(x,y) \mid x \leq x_0, y < y_0\}$;
                                          $D_{L2} = \{(x,y) \mid x > x_0, y \leq y_0\}$;
4. For sorting $D_{H1}$ in Low-distance order:
   While $D_{H1}$ is not empty do the following:
   Begin
   (a) $u = u_0$; Temp_set = {Ø};
   (b) Find the node v with smallest $k(x, y)$ value in $D_{H1}$;
       ( $k(x, y) = |y-y_0|+|x-x_0|$ is distance vector
        from $(x, y)$ to $(x_0, y_0)$ )
      if $k(x_1, y_1) = k(x_2, y_2)$ then
       if $|x_0 - x_1| < |x_0 - x_2|$ then $(x_1, y_1)$ is selected first;
       else $(x_2, y_2)$ is selected first;
   (c) Add node v to Temp_set;   Remove node v from $D_{H1}$;
   (d) $u = v$;
   End.
   $D_{H1}$ = Temp_set;
   Do the same algorithm for sorting $D_{H2}$, $D_{L1}$ and $D_{L2}$;
5. Construct four messages which each one containing one
   of the four subsets ($D_{H1}$, $D_{H2}$, $D_{L1}$ and $D_{L2}$) as part of the
   header.
End.
```

Fig. 3. Message header construction for the proposed multicast routing.

Router Structure: Now, we describe the implementation details of the proposed router. Each input port has a controller for handshaking and an input buffer used for the temporary storage of flits. After receiving the message header, first the routing unit determines which output should be used for routing this message and then the arbiter request for a grant to inject the message to a proper output using the crossbar switch. It also controls the buffer status including empty and full states. A positive rate indicates that the buffer is becoming full while a negative rate reveals that the buffer is becoming empty. The sign is compared to the buffer status to activate a Congestion Flag (CF) [12]. Each input port has a congestion signal (CF) through which it informs its adjacent router about its congestion condition. Therefore, the router which uses that input port for forwarding a message to the next router should consider this router as a congested router and should not send messages to this router until the congestion is over. When a new message reaches the input port, it waits until the previously arrived messages leave the port. Then the new message header is delivered to the routing unit and routed to the appropriate output port. In order to know if the buffer status is critical or not, we should measure the rate at which the data enters and exits the buffer. For this purpose, the circuit shown in Fig. 5 is used. N_{new} is the number of occupied sockets of the input buffer in the current cycle of router clock and N_{old} is the same number but in the previous cycle of router clock. To determine the rate at which the buffer becomes full, the number of filled buffer cells at each rising edge of the router internal clock (N_{new}) is compared to that of the previous rising edge (N_{old}). If $N_{new} > N_{old}$ ($N_{old} > N_{new}$), it will show that the buffer is becoming full (empty). The status signal of the buffer is becomes full when the number of empty cells of the buffer is less than a threshold value. In this case, for warning for the full status, the signal W_Full is activated indicating that most buffer cells are full. This suggest that the congestion condition be traced using the signal W_Full which indicates the filling of the buffer. As shown in Fig. 5, the Congestion Flag (CF) will switch to high when both the W_Full signal and the positive rate for occupying the input buffer sockets are detected. The router employs a routing unit which decodes the header of the message coming from an input port. A minimal path adaptive routing algorithms based on the odd-even turn model which prevents the occurrence of the deadlock [11] is used to determine the output port to which the message should be sent. Therefore, there is no need for implementing virtual channels in the router [12]. In odd-even adaptive routing algorithm there could be more than one minimal output direction to route the messages. In this case the address decoder will choose the direction in which the corresponding downstream router has not raised its congestion flag. For instance, if a message with a given source and destination could be routed to both output p1 (CF=0) and p2 (CF=1), then it will be routed to p1. If p1 and p2 happen to have both their congestion flag raised, the message will be routed to p1.

Consumption Channel Deadlock: In the path based multicast wormhole mechanism, when multiple delivery channels are occupied by one message along the multicast path, cyclic dependencies on the delivery channels may occur [14]. To prevent deadlocks in delivery (consumption) channels, the upper bound of the number of delivery channels required to avoid such deadlocks is equal to 2nv where n is the network dimension and v is the number of virtual channels per input port [14]. As a result, at least two delivery channels are necessary and sufficient for DP, MP, and CP algorithms [14].

5. Results and Discussion

To assess the efficiency of the proposed path based multicast routing algorithm, three other multicast routing algorithms were also implemented. These algorithms include DP, MP, and CP. We have developed a flit level NoC simulator written in C++ capable of calculating the average delay and the power consumption for the message transmission. This simulator can be used for wormhole switching in two-dimensional mesh configuration for the NoC. The simulator inputs include the array size, the router operation frequency, the router algorithm, the link width length, and the traffic type. The simulator can generate different traffic profiles. To calculate the power consumption, we have used Orion library functions[16]. Since the simulator is event driven, the simulation speed is high. For all switches, the data width is set to 32 bits, and each input channel has a buffer (FIFO) size of 3 flits with the congestion threshold set at 60% of the total buffer capacity. The message size was assumed to be 20 flits. The time needed to generate the multicast messages is not considered, because we assumed the multicast messages are generated in the PEs.

Multicast Traffic Profile: The first sets of simulations were performed for a random traffic profile. In these simulations, the processing elements (PE) generates five flit data messages and inject them into the network using the time intervals which are obtained based on the exponential distribution. Two array sizes have been considered 8×8 and 16×16. In the multicast traffic profile, each PE sends a message to a set of destination. A uniform distribution is used to construct the destination set of each multicast message [7]. The number of destinations has been set to 10 and 25. In Fig. 6 and Fig. 7 the average communication delay as a function of the average message injection rate have been plotted. As observed from the results, the DP algorithm achieves better results than the MP algorithm when the number of destinations grows. However, the proposed multicast routing algorithm even in high traffic loads or with a large number of destinations (25 destinations) leads to the lowest delay among all the three multicast routing algorithms.

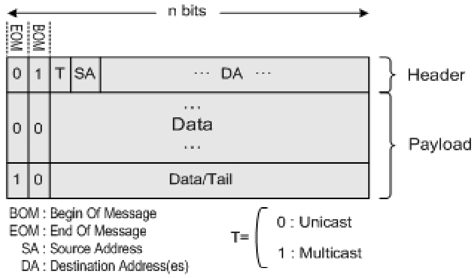


Fig. 4. Multicast message format

Power Dissipation: The power dissipation of DP, MP, CP and the proposed routing algorithms were calculated and compared under the multicast traffic model. The results for the average and maximum power under this traffic are shown in Fig. 8 and Fig. 9, respectively. The average power dissipation of the network with the proposed algorithm by 10 destinations is 25% less than that of the DP algorithm, 3.5% less than that of the MP algorithm, and 33% less than that of the CP algorithm under the multicast traffic model.

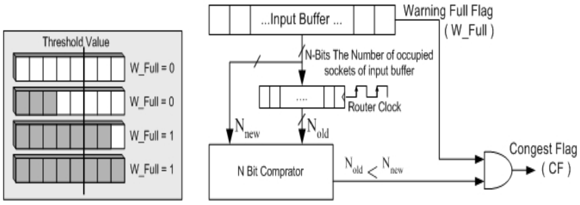


Fig. 5. Congestion detection circuit for the input buffer.

The peak power of the proposed algorithm is 27%, 8%, and 44% less than that of the DP, MP, and CP algorithms, respectively under the multicast traffic model. We can notice that the average power and the peak power compared to other algorithms, is considerably lowered in our proposed algorithm. This is achieved by smoothly distributing the power consumption over the network using the adaptive routing scheme which reduces the number of the hotspots and, hence, lowering the average power and the peak power both.

Hardware Overhead: To evaluate the area overhead of the proposed algorithm, we designed the switches based on the multicast routing schemes and some additional hardware described in router structure. In addition, the destination sorting algorithm depicted in Fig. 3 is implemented in the PEs. The switches were described in VHDL and synthesized with Leonardo-Spectrum ASIC using the SCL 0.25µm standard cell library. For all switches, the data width was set to 32 bits (flit size), and each input channel had a buffer size of 3 flits. In order to achieve better performance/power efficiency, the FIFOs were implemented using registers. The CP, the MP, and the DP multicasting schemes used the same switch structure for implementation. Comparing the area cost of proposed switch with DP (MP, or CP), and XY indicates 2.5% and 8% additional overhead, respectively. The XY is a simple unicast and multicast switch in NoCs.

6. Summary and Conclusion

In this paper, a new path-based multicast algorithm in mesh interconnection networks for NoCs was proposed. In this scheme, three facets have been considered such as utilization of network partitioning, optimized destination ordering, and taking advantage of the odd-even turn model adaptive algorithm for routing both the multicast and unicast messages through the network. Beside, the adaptive routing algorithm used the congestion condition of the input ports to route messages through non-congested paths while it enabled us to distribute the load preventing highly congested area problem. A C++ simulator was used to evaluate the efficiency of the proposed multicast routing algorithm. Under the multicast traffic model and in high message injection rates, the proposed algorithm had the lowest average communication delay in comparison with the DP, the MP, and the CP multicast routing algorithms. It also reduced both the average and the maximum power dissipation of the network compared to those of the DP, MP and CP under the multicast traffic model.

Refrences:

[1] W.O. Cesario, D. L, G. N, Y. P, S.Yoo, L. Gauthier, M. Diaz-Nava, and A.A. Jerraya, “Multiprocessor SoC platforms: a component-based design approach”, in proc. of DATE, pp. 52–63, June 2002.

[2] Z. Lu, B. Yin, A. Jantsch, “Connection-oriented Multicasting in Wormhole-switched Networks on Chip,” in Proc. of ISVLSI, 2006, Germany.

[3] P. K. McKinley, H. Xu, E. Kalns, and L. M. Ni, “CompaSS: Efficient communication services for scalable architectures,” in proc. of supercomputing'92, pp. 478-487, Nov 1992.

[4] H. Xu, P. K. McKinley, E. Kalns, and L. M. Ni, “Efficient implementation of barrier synchronization in wormhole-routed hypercube multicomputers,” Journal of Parallel and Distributed Computing, pp. 172-184, 1992.

[5] K. Li and R. Schaefer, “A Hypercube Shared Virtual Memory,” in Proc. of the Int'l Conf. on Parallel Processing (ICPP'89), pp. 125-132, Aug 1989.

[6] M. Azevedo and D. Blough, “Fault-Tolerant Clock Synchronization of Large Multicomputers via Multistep Interactive Convergence,” in proc of ICDCS’96, pp. 249-257, 1996.

[7] X. Lin, and L. M. Ni, “Multicast Communication in Multicomputer Networks,” IEEE Transactions on Parallel and Distributed Systems, pp. 1105-1117, 1993.

[8] L. M. Ni and P. K. McKinley, “A survey of wormhole routing techniques in direct networks,” IEEE Tran. On Computers, Vol.26, pp. 62-76, Feb. 1993.

[9] J. Hu and R. Marculescu, “DyAD-Smart Routing for Networks-on-Chip,” in Proc. of DAC‘04, pp. 260 - 263, USA.

[10] C. J. Glass and L. M. Ni, “The Turn Model for Adaptive Routing,” in Proc of Computer Architecture, pp. 278-287, May 1992.

[11] G. Chiu, “The Odd-Even Turn Model for Adaptive Routing,” IEEE Tran. On Parallel and Distributed System, pp 729-738, July 2000.

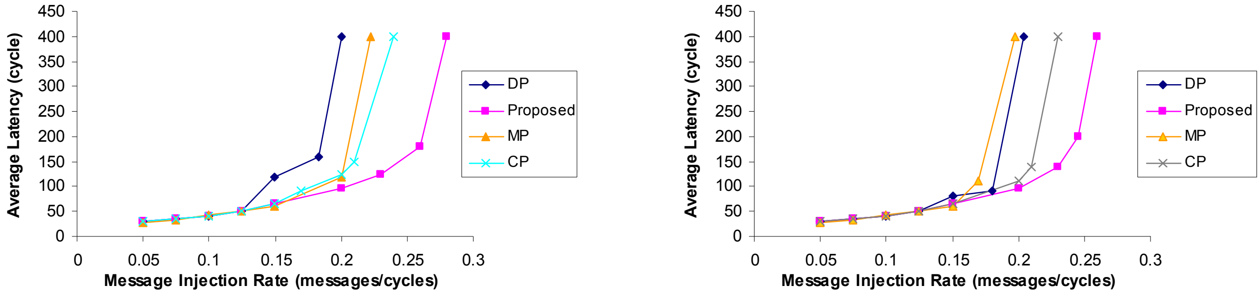
[12] M. Daneshtalab, A. Pedram, M. H. Neishaburi, S. Mohammadi, A. Afzali-Kusha, “Distributing Congestions in NoCs through a Dynamic Routing Algorithm based on Input and Output Selections,” in Proc. of 20th VLSID, pp. 546-550, Jan 2007, India.

[13] M. P. Malumbres, J. Duato, and J. T, "An Efficient Implementation of Tree-Based Multicast Routing for Distributed Shared-Memory Multiprocessors," in Proc. of SPDP, pp. 186-190, 1996.

[14] R. V. Boppana, S. C, C.S R, "Resource deadlock and performance of wormhole multicast routing algorithms," IEEE Transactions on Parallel and Distributed Systems, pp. 535-549, 1998.

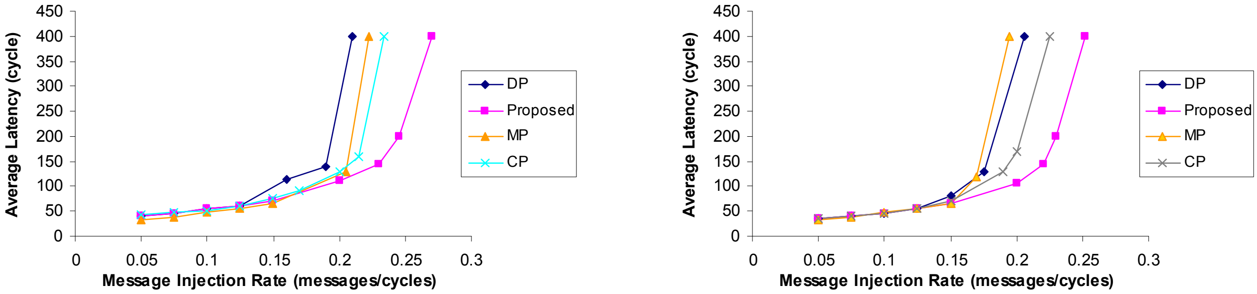
[15] J. Duato, C. Yalamanchili, L. Ni, "Interconnection networks: an engineering approach", Morgan Kaufmann Publishers, 2003.

[16] Wang, X. Zhu, L. Peh, and S. Malik, "Orion: A Power-Performance Simulator for Interconnection Network," In Proc. Hot Interconnection, Stanford, CA, pp. 294 – 305, August 2002.



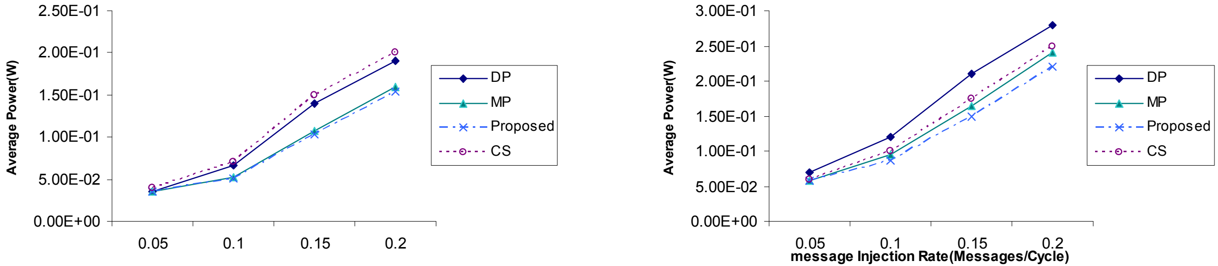
(a) (b)

Fig. 6. Performance under different loads in 8×8 2D-mesh with (a) 10 destinations, (b) 25 destinations.



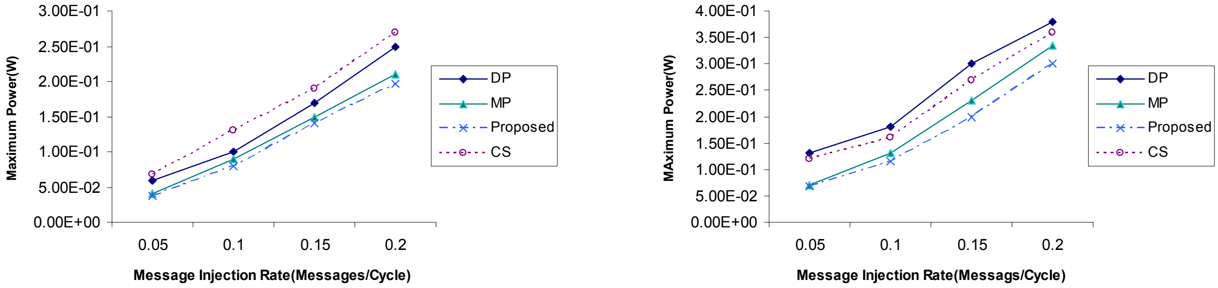
(a) (b)

Fig. 7. Performance under different loads in 16×16 2D-mesh with (a) 10 destinations, (b) 25 destinations.



(a) (b)

Fig. 8. Average power dissipation of the proposed, the DP, the MP and the CP algorithms in 16×16 2D-mesh with (a) 10 destinations and (b) 25 destinations under multicast traffic.



(a) (b)

Fig. 9. Maximum power dissipation of the proposed, the DP, the MP and the CP algorithms in 16×16 2D-mesh with (a) 10 destinations and (b) 25 destinations under multicast traffic.