HAMUM – A Novel Routing Protocol for Unicast and Multicast Traffic in MPSoCs

Masoumeh Ebrahimi, Masoud Daneshtalab, Pasi Liljeberg, Hannu Tenhunen

Department of Information Technology, University of Turku, Turku, Finland
{masebr, masdan, paslil, hanten}@utu.fi

Abstract. *Many parallel applications in MPSoCs take advantage of multicast communication. Several multicast schemes such as path-based, tree-based, and unicast-based have been proposed in interconnection networks. Path-based multicast scheme has been proven to be more efficient than the other schemes in on-chip interconnection network. A new adaptive routing model based on Hamiltonian path for both the multicast and unicast traffics, called Hamiltonian Adaptive Multicast and Unicast Model (HAMUM), is presented. Results obtained in both multicast and mixed traffic models show that the proposed adaptive algorithm for multicast aspect has lower latency and power dissipation compared to previously proposed path-based multicasting algorithms with less than 0.5% hardware overhead. Additionally, for the unicast aspect the proposed adaptive model outperforms the other unicast turn models.*

1. INTRODUCTION

Since the traditional bus-based communication solutions in MPSoC are not useful anymore, new communication architecture is needed. Network on Chip (NoC) has been addressed as a solution for the communication requirement in MPSoCs [1][2]. The Communication in NoC (or MPSoC) can be either unicast or multicast [3]. In the unicast communication a message is sent from a source node (IP or memory) to a single destination node (IP or memory), while in the multicast communication a message is sent from a source node to an arbitrary set of destination nodes. Multicast communications are frequently employed in many applications of MPSoC such as replication [4], barrier synchronization [5], cache coherency in distributed shared-memory architectures [6], and clock synchronization [7].
Multicast routing algorithm can be classified as unicast-based [12][13], tree-based [12][14], and path-based [15][16]. In the unicast-based, the multicast operation is performed by sending a separate copy of a message from the source to every destination or, alternatively, by sending the unicast message to subset of destinations. The drawback of this scheme is the fact that multiple copies of the same message are injected into the network, and then the traffic of the network is increased. Furthermore, each copy of the message suffers from considerable startup latency at the source. In the tree-based multicast approach, a spanning tree is constructed so that the source is indicated as the root and messages are sent down the tree. In this way a message might be replicated at some of the intermediate nodes and forwarded 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 increased network contention. Although such schemes can be used effectively in networks employing store-and-forward and virtual cut-through routing, tree-based routing incurs high congestion in wormhole networks [16]. Some of tree-based multicast routing algorithms such as VCTM [12] have been proposed for on-chip interconnection networks to overcome the tree-based drawbacks. The complexity, and hence, the hardware overhead of this model is hardly depended to the network size which is very critical. A solution to overcome the tree-based disadvantages 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 destinations in the order in which they will 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 data 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 approaches exhibit superior performance characteristics over their unicast-based and tree-based counterparts [9][18].

In this work, we present an adaptive, deadlock-free unicast/multicast wormhole routing algorithm in 2D-mesh NoCs which is inspired by multicomputer networks [8]. The proposed routing model, named HAMUM (Hamiltonian Adaptive Multicast and Unicast Model), is based on Hamiltonian path [9] and like other unicast turn model algorithms such as XY[10], Odd-even[10], DyAD [11] and etc. it restricts the locations where some types of turns can be taken. With these restrictions the algorithm remains deadlock-free and does not require virtual channels. Although the adaptive turn model routing algorithms are applicable just for the unicast approach, our model accomplishes the adaptivity to the both unicast and multicast approaches. Additionally, the degree of the routing adaptiveness provided by our model is higher than the adaptiveness of unicast adaptive turn models such as Odd-Even. Experimental results with multicast and synthetic mixed traffic profiles show that power and performance can be improved by using the proposed adaptive model in traditional path-based multicast algorithms such as Multi

Path, and Column Path. Besides, in the unicast approach our adaptive proposed model outperforms the traditional adaptive routing models. The chip area overhead of the proposed scheme is negligible, less than 0.5%. The paper is organized as follows. In Section 2, a brief review of the traditional path-based multicast algorithms is presented. In Section 3, the proposed adaptive path-based model is 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. HAMILTONIAN PATH-BASED STRUCTURE

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 (x_i, y_i) and (x_j, y_j) are connected by a communication channel if and only if $|x_i - x_j| + |y_i - y_j| = 1$.

The path-based routing algorithm is established as the Hamiltonian path algorithm [8].In this method an undirected Hamiltonian path of the network is constructed; A Hamiltonian path visits every node in a graph exactly once [19]. In this algorithm, for each node in an $m \times n$ 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 shown in Fig. 1, two directed Hamiltonian paths (or two subnetworks) are constructed by the labeling [8]. 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 Hamiltonian path (the dashed lines in the Fig. 1) could be used in appropriate directions. The proposed adaptive model designed for both unicast and multicast messages, uses the Hamiltonian path strategy. The Multi-Path (MP) [8] and Column-Path (CP) [9] algorithms are the most important path-based routing methods that use the Hamiltonian path strategy, which are described as follow:

MP Multicast Routing: In the Multi-Path (MP) 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 of the source node. Thus, multicast messages from the 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. To reduce the path lengths, D_H and D_L 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 subset contains the remaining nodes in D_H. The set D_L is partitioned in a similar way. Hence, all destinations of multicast message are grouped into four disjointed subnetworks. Consider the example illustrated in Fig. 2(a) for a 8×8 mesh network where node 27 (3, 4) sends its multicast messages to destinations 0, 1, 7, 8, 9, 19, 26, 31, 32, 37, 50, 55, 57, 59, 62, and 63. 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 31, 32, 37, 50, 55, 57, 59, 62, and 63 in sequence and the second one (D_L) has the remaining destinations that could be reached using the H_l subnetwork which are 0, 1, 7, 8, 9, 19, and 26. As exhibited in Fig. 2, D_H is divided into two subsets, which are D_{H1}= {31, 32, 50, 62, 63} and D_{H2}= {37, 55, 57, 59}. In the same way D_L is divided into two subsets, with D_{L1}= {0, 1, 19} and D_{L2}= {7, 8, 9, 26}. The multi-path is deadlock-free and could be used for unicast and multicast routing simultaneously.

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 the source, then one copy of the message is sent to service all of those destinations. Similarly, if a column has one or more destinations in the rows below the source, then another copy of the message is sent to service all of 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 0, 1, 7, 8, 9, 19, 26, 31, 32, 37, 50, 55, 57, 59, 62, and 63 from the source node 27 using the Column-Path routing algorithm based on the Hamiltonian path is shown in Fig. 2(c). Thirteen copies of the message are used to achieve the desired multicast operation. Though destinations 1 and 62 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 used in this scheme is based on the XY routing algorithm. Therefore, the CP routing algorithm is compatible with the unicast routing method and it is deadlock-free and livelock-free [9].

3. THE HAMUM MODEL

The former path-based routing models such as MP [8] and CP [9] algorithms that have been described, route the unicast and multicast messages by using deterministic routing algorithms. Therefore, the network performance is degraded by these models. The proposed minimal adaptive scheme takes the place of the deterministic model in the path-based routing algorithms to route both of the unicast and multicast messages through the destination(s).

For breaking all of the cycles in the proposed adaptive scheme, similar to the odd-even model, the locations where certain turns can be taken are restricted so that deadlock will be avoided. The rules regulating the proposed scheme are

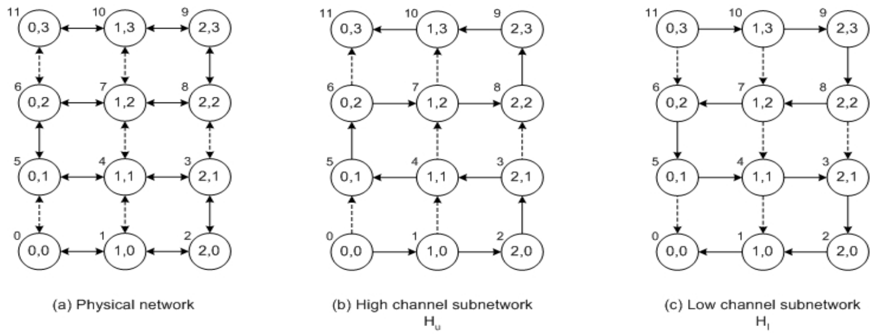

Fig. 1. (a) A 3×4 mesh physical network with the label assignment and the corresponding (b) high channel and (c) low channel networks. The solid lines indicate the Hamiltonian path and dashed lines indicate the links that could be used to reduce path length in routing.

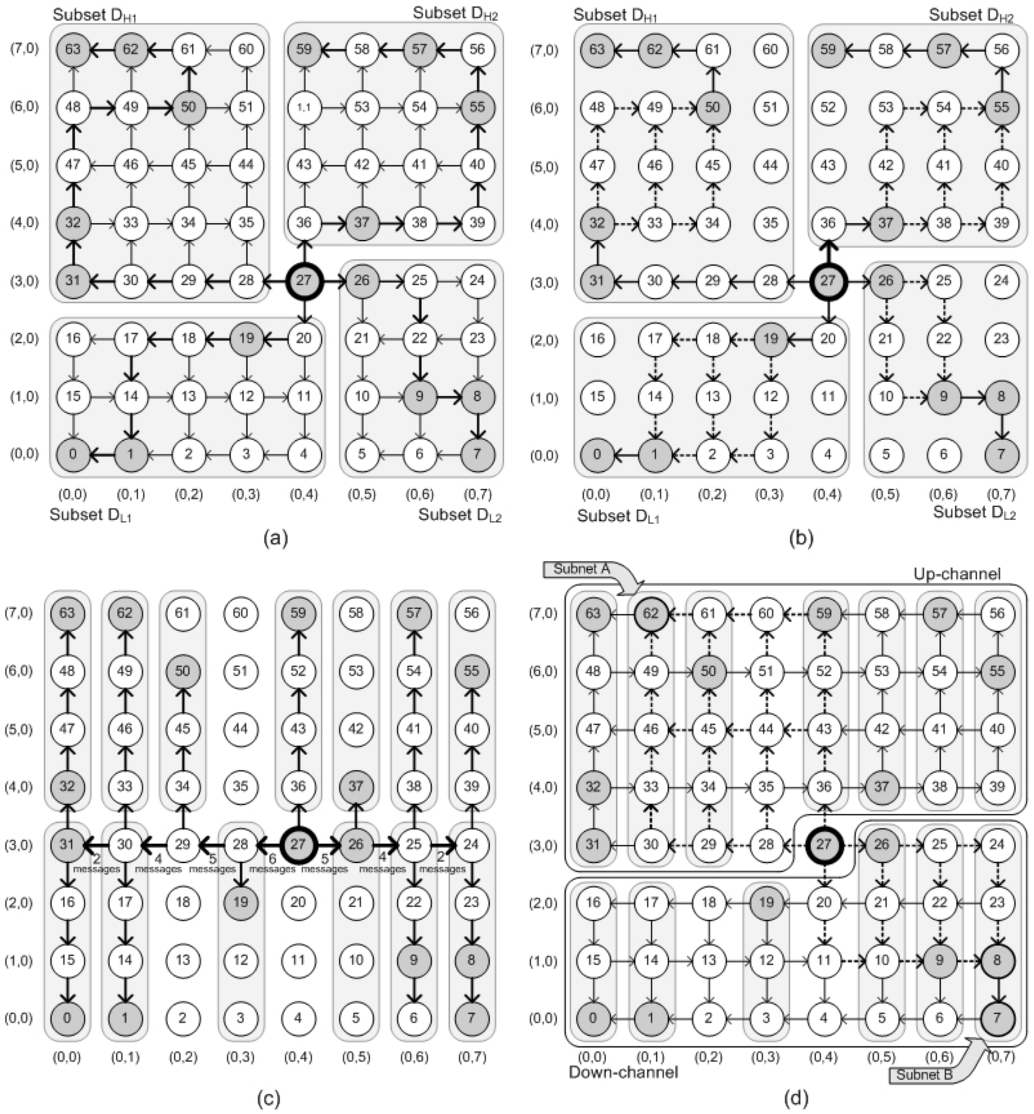

Fig. 2. Examples of (a) Multi-path (MP), (b) Adaptive Multi-Path (AMP), (c) Column-Path (CP), and (d) Adaptive Column-Path (ACP) multicast routings from the node 27 (to subsets A, and B for column path example). The unused links are not indicated.

categorized in the high channel subnetwork and the low channel subnetwork as follows:

For the *high channel subnetwork*:

Rule1: NW turn is not allowed in even rows.

Rule2: NE turn is not allowed in odd rows.

For the *low channel subnetwork*:

Rule1: SE turn is not allowed in even rows.

Rule2: SW turn is not allowed in odd rows.

Notice that the message will be forwarded to the destination as in the deterministic Hamiltonian algorithm, when the current node is located one row to the south (north) of the destination row in the high channel network (low channel network). Inasmuch as the rules keep the messages traveling through the Hamiltonian paths, it prevents the occurrence of deadlock [9]. In addition, both minimal and non-minimal paths are possible with the proposed adaptive path-based model. However, our implementation is based on minimal paths and does not support the non-minimal paths.

3.1. Multicast Aspect of the HAMUM Model

Now we describe how the proposed adaptive model affects the path-based multicast routing algorithms.

AMP, Adaptive MP, is the adaptive model of the MP algorithm after the proposed adaptive model (HAMUM) is applied in the MP algorithm. Consider the example used for MP in Fig. 2(b).The multicast message can be forwarded in three different ways from the node 37 through the node 55 (32 through 50, 19 through 1, and 26 through 9).

ACP, stood for the Adaptive CP, is the adaptive method of the original CP by taking advantage of the proposed adaptive model. To indicate how the adaptive scheme affects the CP algorithm, as illustrated in Fig. 2(d), again thirteen copies of the multicast message must be used to achieve the desired multicast operation. But in this figure for simplicity, we only consider two subsets A and B. Due to utilizing the proposed adaptive scheme in the CP, each multicast message can be delivered to its subset through different paths indicated by dashed lines in the example of Fig. 2.

3.2. Unicast Aspect of the HAMUM Model

Based on the proposed model, any intermediate node must first determine the set of directions where a packet may be forwarded for the next hop based on Rule 1 and Rule 2.

As mentioned earlier, according to the source and destination labels, the routing may take place in high or low channel. Consider the case where the destination of a message is to the west of its source through the high channel network. If the current node is in the odd row, the router can route the message to the west or north direction because of the Hamiltonian high channel network strategy. If the current node is in the even row, at first the message should be routed to the north direction (to reach the odd row), and then, because the message reaches to the odd row it could be routed even to the west or north. Note that in the high channel subnetwork, using Hamiltonian path, the packet can choose west or north direction in odd rows and east or north direction in even rows. Additionally, if the current node is located one row to the south of the destination row in the high channel network, the message will be routed to the west direction if the current node is in the odd row, and if the current node is in the even row, the packet will be routed to the north direction. In Fig. 3, all the possible minimal routing paths for one unicast message in the 5x5 2D-mesh have been exhibited. That is, at least one minimal path always can be selected by the proposed model for any source and destination pair.

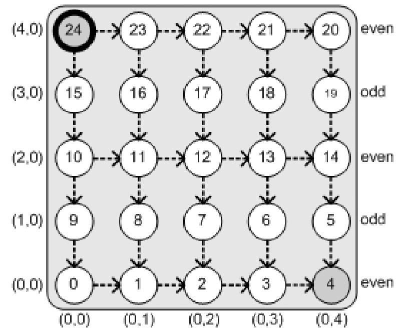

Fig. 3. All of the possible minimal paths from the source node 24 to the destination node 4 of the proposed model for unicast messages.

4. Hardware Implementation

In this work, due to scalability, cross-section bandwidth, and the fixed degree of nodes, we use an $n \times n$ network of interconnected tiles with a mesh topology [12][21]. 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 the 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 [17].

4.1. Message Format

The multicast message format is shown in the Fig. 4. It includes one or several header flits and a parametric number of payload flits. The number of flits depends on the number of destinations and the flit width in the network. 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 messages: unicast (*T=0*) and multicast (T=1), indicated by T. The specific address of the source node, the pointer counter, and the destination node address(es) are placed in the last field of the header, respectively, and the content of the message is located in the rest of the flits (Payload). The pointer indicates the address of the next destination in the header flit, and the *MID* used for message ordering.

4.2. SWITCH STRUCTURE

Each input port has a controller for handshaking and an input buffer used for the temporary storage of flits. The wormhole switching method implemented in the controller unit, is based on on/off flow control mechanism [22]. After receiving the message header, first the routing unit determines which output should be used for routing this message and then the arbiter requests for a grant to inject the message to a proper output using a crossbar switch. The router has the crossbar which establishes the connection path from an input port to an output port. Since the crossbar can only serve a single output port at a time, it uses the arbitration among simultaneous input requests to access the same output port. When a new message reaches the input port, it waits until the previously arrived messages leave the port. Then the header of the new message is delivered to the routing unit and routed to the appropriate output port. The Congestion Flag *(CF)* of the buffer becomes active when the number of empty cells of the buffer is less than a threshold value. In this case, warning for the full status, the signal *CF*, is activated indicating that most buffer cells are occupied. Each input port has a *CF* through which it informs its adjacent routers 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 one (congestion area or hotspot) and should not send messages to this router until the congestion condition is over.

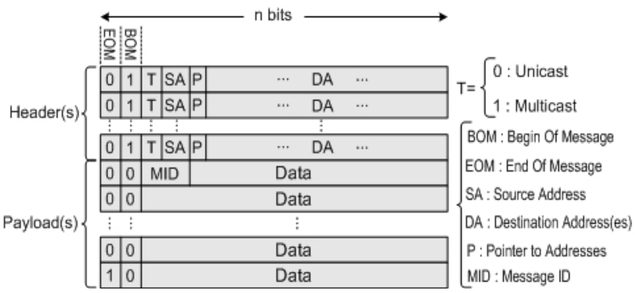

Fig. 4. Multicast message format

4.3. 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 [9][15]. 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 [9][15]. As a result, at least two delivery channels are necessary and sufficient for MP and CP algorithms [15].

4.4. HEADER PROCESSING MECHANISM

The router employs a routing unit which decodes the header of messages coming from an input port. If the header belongs to a unicast message (T=0), the minimal path adaptive routing algorithms is used to determine the output port to which the message should be sent. In the proposed adaptive routing algorithm there could be more than one minimal output directions where to route messages. In this case the address decoder will choose the direction where 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*. On the other hand, if the header type is a multicast message (*T=1*), the routing unit fetches the destination address from where the pointer in the header points. Afterward, the routing unit increases the pointer value of the header, and if it overflowed, the routing unit would remove the corresponding flit header from the message. In a word, whenever a destination address is fetched from the header, the pointer value will be increased. After fetching the destination address from the header, if the destination address is the current node, the routing unit will request the local output port. Meanwhile, the routing unit fetches the next destination address from the header and runs the adaptive routing procedure to determine the output port(s) corresponding to the next destination address.

4.5. OUT-OF-ORDER HANDLING

As a result of exploiting the proposed adaptive routing algorithm, messages on the two different paths can be out-of-order. Hence, we need a mechanism to re-order the messages at the destination. The received message at the destination node will be stored in a temporary memory. The address memory of the received message is generated by combining the source address and the message identifier of the corresponding message. After all parts of the data are delivered to the temporary memory, the origin data will be delivered to the main memory. The temporary memory can handle several data flows concurrently.

5. RESULTS AND DISCUSSION

To assess the efficiency of the proposed adaptive model, two multicast routing algorithms were implemented. These algorithms include MP and CP. We have developed a synthesizable wormhole NoC simulator implemented in VHDL to assess the efficiency of the proposed adaptive method. This simulator can be used for wormhole switching in two dimensional mesh configuration. The simulator inputs include the array size, the routing algorithm, the link width, buffer size, and the traffic type. The simulator can generate different traffic profiles. To calculate the power consumption, we have used power compiler. For all switches, the data width was set to 32 bits, and each input channel has a buffer (FIFO) size of 12 flits with the congestion threshold set at 75% of the total buffer capacity. The message size was assumed to be 16 flits. For the performance metric, we use the multicast latency defined as the number of cycles between the initiation of multicast message operation and the time when the tail of the multicast message reaches all the destinations.

5.1. Multicast Traffic Profile

The first set of simulations was performed for a random traffic profile. The array size was considered to be 8×8. In the multicast traffic profile, each PE (Processing Element) sends a message to a set of destinations. A uniform distribution is used to construct the destination set of each multicast message [8]. The number of destinations has been set to 10 and 25. The average communication delay as a function of the average flit injection rate has been shown in Fig. 5. As observed from the results, the proposed adaptive mechanism which has been applied to MP and CP even in high traffic loads or with a large number of destinations (25 destinations) leads to lower delay.

5.2. Unicast and Multicast (Mixed) Traffic Profile

In this set of simulation, we have employed a mixture of unicast and multicast traffic, where 80% of injected messages are unicast messages and the remaining 20% are multicast messages. This pattern may be representative of the traffic in a distributed shared-memory multiprocessor where updates and invalidation produce multicast messages and cache misses are served by unicast messages [9][14]. The unicast messages are also routed using the proposed adaptive scheme. Uniform traffic [10] and hotspot [10] are two different traffic profiles that have been taken into account for unicast traffic generation. In the uniform traffic profile, each PE sends a message to any other PE in an equal probability. This is determined randomly using a uniform distribution. Under the hotspot traffic pattern, one or more nodes are chosen as hotspots receiving an extra portion of the traffic in addition to the regular uniform traffic. In Fig. 6 the average communication latency of different algorithms under the uniform traffic model for unicast traffic is shown. As depicted in these figures, for this traffic, the adaptive routing algorithms perform better. Under the hotspot traffic model, given a hotspot percentage of h, a newly generated message is directed to each hotspot node with an additional h percent probability. We simulate hotspot traffic with a single hotspot node. The hotspot node is chosen to be node *(4, 4)* in the 8×8 2D-Mesh. Fig. 7 shows the multicast routing performance with h = 10%. As the figure shows, the adaptive proposed routing algorithm outperforms the traditional algorithms.

5.3. Unicast Traffic Profile

For appraising the unicast efficiency of the proposed routing algorithm, three unicast routing algorithms were implemented. These algorithms include XY [10], Odd-Even [10], and DyAD [11]. The hotspot traffic profile, where 100% of injected messages are unicast messages has been considered. The node *(4, 4),* and node *(8, 8)* are chosen as the hotspot nodes with h=15% in the 8x8, and 14x14 2D-Mesh respectively. Fig. 8 shows the simulation results for the hotspot traffic. As depicted, the HAMUM algorithm outperforms the other unicast routing algorithms. Particularly, when the network size is increased our algorithm is superior to all of the others. This can be seen in Fig. 8 (b).

5.4. Hardware Overhead

To evaluate the area overhead of the proposed algorithm, the switches were synthesized Synopsys D.C. using the TSMC 0.09μm standard cell library. In addition, the destination sorting algorithms are included in the hardware overhead. For all switches, the data width was set to 32 bits (flit size), and each input channel had a buffer size of 12 flits. As discussed in section 4, for the MP, and CP switches we use two delivery channels. In order to achieve better performance/power efficiency, the FIFOs were implemented using registers. The CP and MP multicasting schemes used the same switch structure for the implementation, but their sorting mechanisms uses different number of registers. Comparing the area cost indicates that the hardware overhead of implementing the proposed adaptive scheme in both the MP and CP switches is less than 0.5% and that can be considered negligible.

5.5. Power Dissipation

The power dissipation of MP, CP, AMP, and ACP were calculated and compared under the multicast traffic model with 25 destinations using Synopsys PrimePower. The typical clock of 1 Giga Hertz is applied to the in 8x8 2D-mesh network. The results for the average and maximum power under this traffic are shown in Fig. 9 (a) and (b) respectively. As the results reveal, the average power dissipation of the network with the ACP algorithm is 5% less than that of the CP algorithm and the average power dissipation of the AMP is 3.5% less than that of the MP algorithm. The results of Fig. 9 (b) indicate that the peak power of the ACP and MCP algorithms is 15% and 11% less than that of the CP and MP algorithms, respectively under the multicast traffic model. We can notice that the average power and the peak power of the proposed adaptive model are lower. This is achieved by smoothly distributing the power consumption over the network using the adaptive routing scheme which reduces the number of hotspots and, hence, lowering the average and peak power both.

6. Summary and Conclusion

In this paper, a new adaptive model based on Hamiltonian path in mesh interconnection networks was proposed. In this scheme, three facets have been considered as the utilization of network partitioning, and taking advantage of the proposed adaptive model for routing both the multicast and unicast messages through the network.

Additionally, 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. Under the multicast and mixed traffic models the proposed adaptive model had lower average communication delay in comparison with the traditional MP, and the CP multicast routing algorithms.

ACKNOWLEDGMENT

The authors wish to acknowledge Nokia Foundation for the partial financial support during the course of this research.

REFERENCES

[1] W. O. Cesario, D. Lyonnard, Y. Paviot, S. Yoo, M. Gauthier, M. Diaz-Nava, and A. A. Jerraya, “Multiprocessor SoC platforms: a component-based design approach,” in proc. int. conf. IEEE Design and Test of Computers, Paris, France, 2002, pp. 52–63

[2] E. Rijpkema, K.G.W. Goossens, A. Radulescu, J. Dielissen, J. van Meerbergen, P. Wielage, and E. Waterlander, “Trade offs in the design of a router with both guaranteed and best-effort services for networks on chip,” in proc. int. conf. Design Automation and Test Conference in Europe, Munich, Germany, 2003, pp. 350–355

[3] Z. Lu, B. Yin, A. Jantsch, “Connection-oriented Multicasting in Wormhole-switched Networks on Chip,” in proc. int. conf. ISVLSI, Germany, 2006, pp. 205-2110

[4] P. K. McKinley, H. Xu, E. Kalns, and L. M. Ni, “CompaSS: Efficient communication services for scalable architectures,” in proc. int. conf. supercomputing, Minnesota, USA, 1992, pp. 478-487

[5] 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, 1992, 16, pp. 172-184

[6] K. Li, and R. Schaefer, “A Hypercube Shared Virtual Memory,” in proc. int. conf. ICPP, USA, 1989, pp. 125-132

[7] M. Azevedo, and D. Blough, “Fault-Tolerant Clock Synchronization of Large Multicomputers via Multistep Interactive Convergence,” in proc. int. conf. ICDCS, Hong Kong, 1996, pp. 249-257

[8] X. Lin, and L. M. Ni, “Multicast Communication in Multicomputer Networks,” IEEE Transactions on Parallel and Distributed Systems, 1993, 4, pp. 1105-1117

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

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

[11] J. C. Hu, and R. Marculescu, “DyAD - smart routing for networks-on-chip,” in Proc. of Design Automation Conference, pp. 260–263, 2004.

[12] N. E. Jerger, L. S. Peh, and M. H. Lipasti, “Virtual Circuit Tree Multicasting: A Case for On-Chip Hardware Multicast Support”. proc. int. conf. Computer Architecture, China, 2008, pp. 229-240

[13] P. McKinely, H. Xu, A. H. Esfahanian, and L. Ni, “Unicast-based multicast communication in wormhole-routed networks,” IEEE Trans. of Parallel and Distributed Systems, 1994, 5, pp. 1252-1265

[14] M. Malumbres, J. Duato, and J. Torrellas, “An Efficient Implementation of Tree-Based Multicast Routing for Distributed Shared-Memory Multiprocessors,” proc. int. conf. SPDP, New Orlean, USA, 1996, pp. 186-190

[15] E. A. Carara, F. G. Moraes, "Deadlock-Free Multicast Routing Algorithm for Wormhole-Switched Mesh Networks-on-Chip," in Prof. of ISVLSI, pp.341-346, 2008

[16] X. Lin, and L. M., Ni, “Multicast Communication in Multicomputer Networks,” IEEE Transactions on Parallel and Distributed Systems, 1993, 4, pp. 1105-1117

[17] J. Duato, Ni, C., L. Yalamanchili, “Interconnection networks: an engineering approach” (Morgan Kaufmann Publishers, 2003)

[18] A. Al-Dubai, I. Romdhani, “A Performance Study of Path Based Multicast Communication Algorithms,” in proc. int. conf. PARELEC, Bialystok, Poland, 2006, pp. 245-250

[19] F. Harary, ‘Graph Theory’ (Readings, Massachusetts: Addson-Wesley, 1972)

[20] C.J. Glass and L.M. Ni, “The Turn Model for Adaptive Routing,” in Proc. 19th Ann. Int'l Symp. Computer Architecture, pp. 278±287, May 1992.

[21] J. Liang, S. Swaminathan, and R. Tessier, “aSOC: a scalable, single-chip communication architectures”, in proc. int. conf. PACT, Oregon, USA, 2000, pp. 37–46

[22] N. Concer, M. Petracca, L. Carloni, “Distributed Flit-Buffer Flow Control for Networks-on-Chip”, in proc. int. conf. CODES+ISSS, Georgia, USA, 2008, pp. 215-220

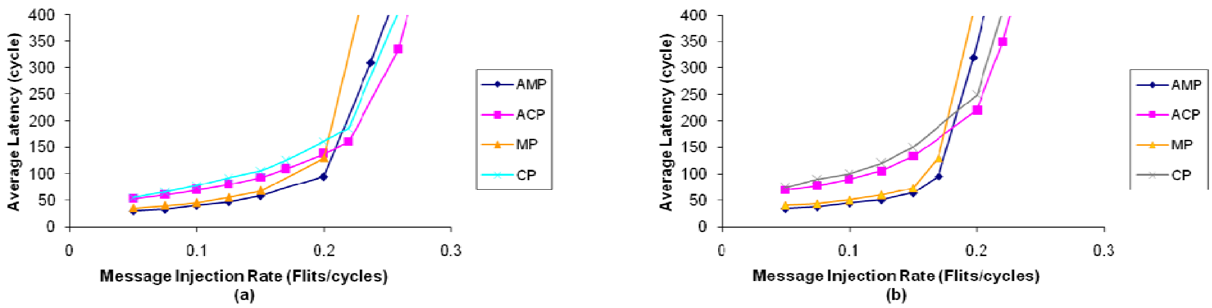

Fig. 5. Performance with different loads in 8×8 2D-mesh with (a) 10 destinations, (b) 25 destinations.

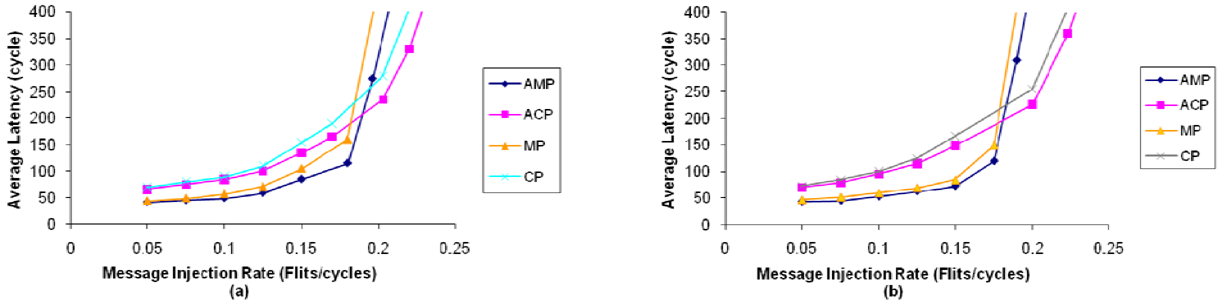

Fig. 6. Performance with different loads in 8×8 2D-mesh with (a) 10 destinations, (b) 25 destinations under mixed traffic (20% multicast and 80% unicast). Unicast traffic is based on the uniform traffic model.

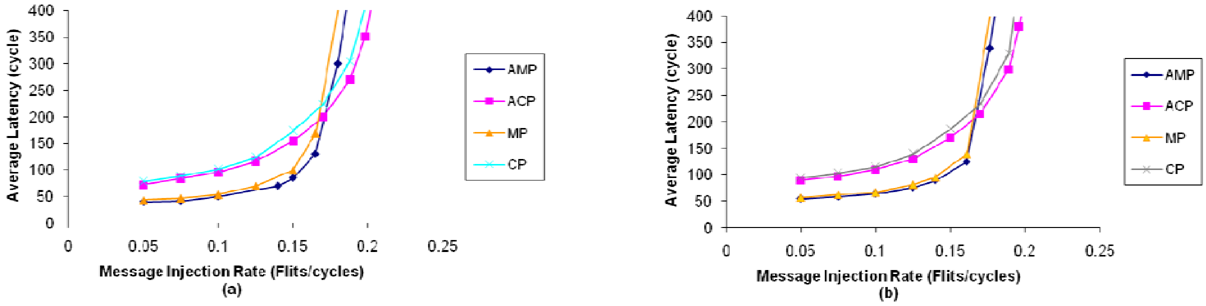

Fig. 7. Performance with different loads in 8×8 2D-mesh with (a) 10 destinations, (b) 25 destinations under mixed traffic (20% multicast and 80% unicast). Unicast traffic is based on the hotspot traffic model with a single hotspot node (4, 4), and h=10%.

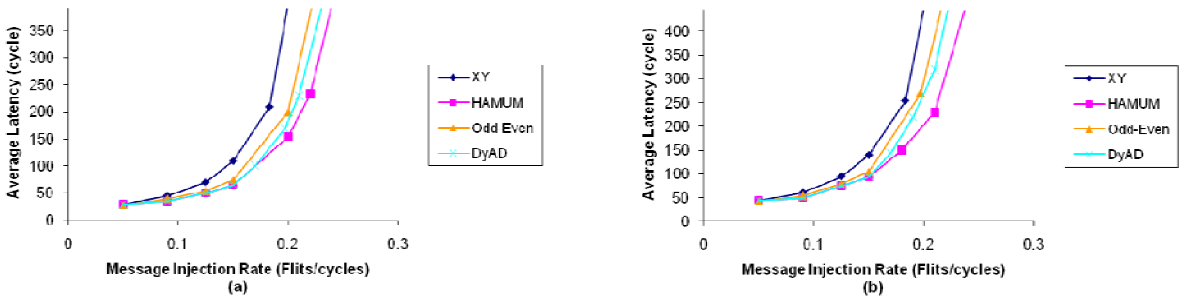

Fig. 8. Performance with different loads in (a) 8×8 2D-mesh, and (b) 14x14 2D-mesh under the hotspot traffic model with a single hotspot node, and h=10 percent.

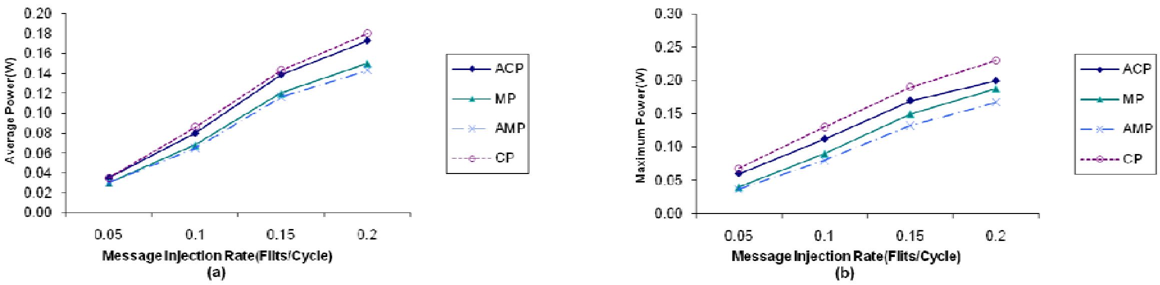

Fig. 9. (a) Average and (b) Maximum power dissipation of the MP, CP, AMP, ACP algorithms in 8×8 2D-mesh with 25 destinations under multicast traffic.