

Input-Output Selection Based Router for Networks-on-Chip

Masoud Daneshtalab², Masoumeh Ebrahimi^{1,2}, Pasi Liljeberg², Juha Plosila², Hannu Tenhunen^{1,2}

¹Turku Centre for Computer Science (TUCS), Joulahaisenkatu 3-5, 20520 Turku, Finland

²Department of Information Technology, University of Turku, Finland

{ masdan, masebr, pakrli, juplos, hatenhu }@utu.fi

Abstract - In this paper, we propose a novel on-chip router architecture for avoiding congested areas in regular two-dimensional on-chip networks. This architecture takes advantage of an efficient adaptive routing model based on the Hamiltonian path for both the multicast and unicast traffic. The output selection of the proposed architecture is based on the congestion condition of neighboring routers and the input selection is based on the Weighted Round Robin mechanism which allows packets to be serviced from each input port according to its congestion level. The simulation results show that in multicast, unicast, and mixed traffic profiles the proposed model has lower average delays and lower average and peak power compared to previously proposed models.

I. INTRODUCTION

Since the traditional bus-based communication solutions in Multi-Processor System-on-Chips (MPSoCs) are not useful anymore, new communication architecture is needed. Network on Chip (NoC) has been addressed as a solution for the communication requirement for MPSoCs [1]. The performance and efficiency of NoC's largely depend on the underlying routing technique which decides the direction a packet should be sent. In the routing process, the *output selection* and the *input selection* are two key components of the router architecture.

The output selection, which is performed using a *routing algorithm*, determines which of the multiple output channels should be chosen for a packet arrived from an input channel. The routing algorithms could be classified as deterministic and adaptive [2]. In deterministic routing models, the path between a source and a destination of a packet is determined by the source and the destination themselves and the current traffic status of the network is not considered. In adaptive algorithms, however, the path between a source and a destination is determined node by node depending on the network status as packets move toward the destination. The adaptive nature of this type of routing algorithms makes them very attractive [2], e.g., Odd-Even [3], DyAD [4] and HAMUM [5] are adaptive routing algorithms and XY [6] is a deterministic routing algorithm in NoC. Communication in NoC (or MPSoC) can be either unicast (one-to-one) or multicast (one-to-many) [7][8]. In unicast communication, a message is sent from a source node to a single destination node, while in multicast communication a message is sent from a source node to an arbitrary set of destinations. Multicast communication is employed in many MPSoC applications, e.g., replication, barrier synchronization, cache coherency in distributed shared-memory architectures, and clock synchronization [8]. Although Multicast communication can be implemented by multiple unicast communications, this alternative method produces too much unnecessary traffic and probably latency and congestion in the network will be increased [8]. Multicast routing algorithms can be classified as unicast-based, tree-

based, and path-based [8]. It has been proven that in on-chip networks, the path-based multicast method is more efficient than the other multicast methods [8][10][11]. In the path-based method, a source node prepares a message for delivery to a set of destinations by first sorting the addresses of destinations in order in which they are to be delivered, and then placing this sorted list in the header of the message.

The input selection chooses one of input channels to get access to the output channel. This is done by an arbitration process. The arbiter could follow either non-priority or priority scheme [4] [13]. In the non-priority scheme when there are multiple input port requests for the same available output port, the arbiter uses the First-Come-First-Served (FCFS) [4][14], or Round-Robin (RR) [4][12] policy to grant access to one input port, such that the starvation on a particular port is avoided (fair). On the other hand, in the priority method when there are multiple input port requests for the same available output port, the arbiter would grant access to the input port request which has the highest priority level [13]. The problem with the priority method is that the starvation could occur (unfair).

In this paper, a novel router architecture, which utilizes both input and output selections, is proposed. The output selection uses an efficient adaptive wormhole routing algorithm, named HAMUM [5]. HAMUM is a Hamiltonian path-based routing model which routes both unicast and multicast traffic adaptively in mesh-based on-chip networks. The input selection, profits from the advantages of both priority (unfair) [13] and non-priority (fair) [4][12][10] arbitration policies. This scheme is called Weighted Round Robin (WRR)[15]. WRR allows a weight to be assigned to each input port. This weight specifies the number of packets to be transmitted whenever the router services that input port. The weight of each input port is proportional to the Congestion Level (CL) of upstream routers. CL is produced by Congestion Aware Routing Selection (CARS) which is part of the router structure. The CL relates to the load level of the router and is sent to immediate neighbors (upstream router) in all directions. The paper is organized as follows. In section II, we review the related work while in section III, the background is described and the router architecture is presented in section IV. The results are discussed in section V and the summary and conclusion are given in section VI.

II. RELATED WORKS

Several routing algorithms for improving the performance of routers in on-chip networks have been proposed. The Turn model is a wormhole routing algorithm that is deadlock and live-lock free [8]. This model has been later utilized to develop an odd-even adaptive routing algorithm for meshes without virtual channels [3]. The routing algorithms proposed in [4][16], perform output selection and wormhole routing based

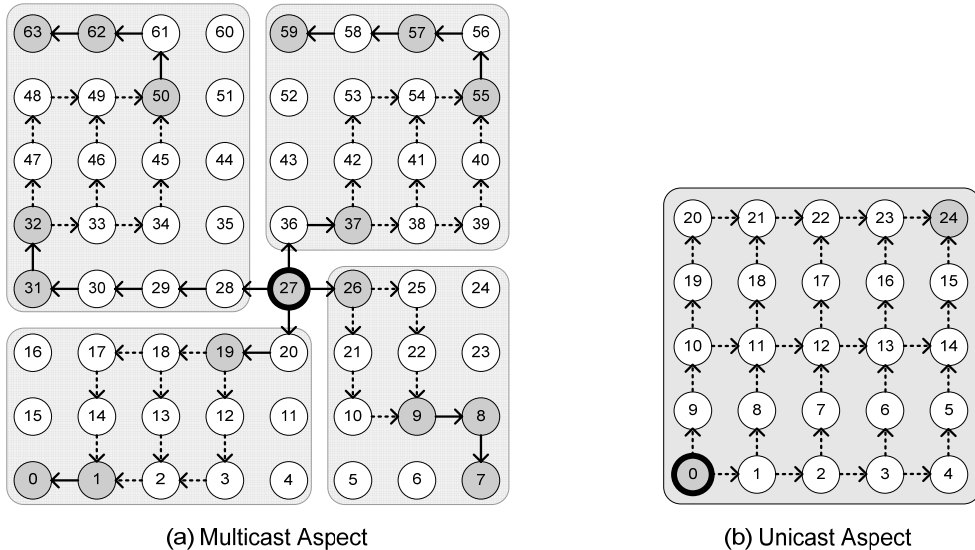


Fig. 1. Examples of (a) multicast aspect and (b) unicast aspect of HAMUM.

on the congestion condition of the neighbor routers. It causes packets to be forwarded to routers with lower traffic load. If turn model algorithms are adopted to route multicast packets, some forbidden turns might be occurred [8][10]. To cope with forbidden turns the absorb-and-retransmission mechanism, which degrades the performance, is required [8][10]. In [8] authors utilized the odd-even routing algorithm to route multicast packets. The more frequently forbidden turns occur the more performance is degraded. HAMUM has been recently proposed to support both unicast and multicast traffic adaptively [5]. Not only the adaptivity of the HAMUM routing algorithm is higher than the adaptivity of Odd-Even for the unicast traffic, but also for the multicast traffic the adaptivity of HAMUM is higher than conventional multicast routing algorithms [5].

The focus of the aforementioned routing schemes is on the output selection realm. Routing techniques concerning the input selection, applied in NoCs, are FCFS, RR, and the contention-aware input selection (CAIS) [13]. Both FCFS and RR are fair to all channels but do not consider the traffic condition of the input channels. In CAIS, the busiest input channel obtains the highest priority to access the output channel. The input channel is given priority proportional to the number of requests arrived from the upstream routers. Thus, the traffic can be kept flowing in busy channels to avoid the network congestion. However, this model increases the possibility of the starvation. In this paper, a router which uses both input and output selections is presented.

III. BACKGROUND

The HAMUM routing algorithm is based on the Hamiltonian path-based model in mesh-based on-chip networks with wormhole switching technique [5]. In the Hamiltonian path-based approach every node in a graph is visited exactly once [10]. The former path-based routing models such as Multi-Path (MP) and Column-Path (CP) algorithms [10] route the unicast and multicast messages by using deterministic routing algorithms. Therefore, the network performance has been degraded by these models. Hence, these path-based routing algorithms can be replaced by HAMUM, a

minimal adaptive scheme to route both unicast and multicast traffics adaptively through the destination(s). For breaking all of cycles in HAMUM, similar to the odd-even model, the locations at where certain turns can be taken are restricted so that deadlock can be avoided.

Fig. 1(a) shows how HAMUM brings adaptivity to Multi-Path (MP), a conventional path-based multicast routing algorithm. In MP routing algorithm the destination 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 high-channel subnetwork and to the destination nodes in D_L using low-channel subnetwork [5][10]. 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 a multicast message are grouped into four disjoint. Consider the example for a 8×8 mesh network where node 27 send its multicast messages to destinations 0, 1, 7, 8, 9, 19, 26, 31, 32, 37, 50, 55, 57, 59, 62, and 63. As exhibited in Fig. 1(a), 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 $D_{L2} = \{7, 8, 9, 26\}$. Consider the example in Fig. 1(a), 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) by the HAMUM routing algorithm.

The adaptiveness of the unicast aspect of HAMUM is depicted in Fig. 1(b). Based on the proposed model, any intermediate node must first determine the set of directions toward which a packet may be forwarded for the next hop based on the rules described in [5]. 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 unicast message is to the east of its source through the high channel network. All the possible minimal routing paths for

one unicast message in the 5x5 2D-mesh are exhibited in Fig. 1(b). It has been revealed that HAMUM outperforms other adaptive routing algorithm under unicast traffics [5].

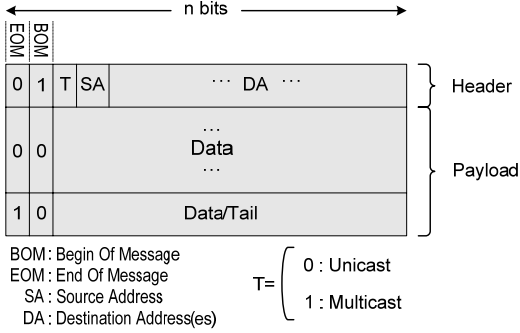


Fig. 2. Message format.

IV. ROUTER ARCHITECTURE

In this architecture, we attempt to spread congestion areas and improve the performance of the network through the simultaneous use of adaptive input and output selection routing algorithms. The output selection of this router adopts HAMUM based on congestion condition of neighbors' routers. In our proposed router the input selection exploits the WRR policy which makes the routing algorithm non vulnerable to starvation. Also, WRR would increase the performance of the algorithm by probing the traffic condition.

Message Format: The message format is shown in Fig. 2. 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 n^{th} bit is the EOM (End Of Message) sign and the $(n-1)^{\text{th}}$ bit is the BOM (Begin Of Message) sign. In the header, the third field T is used to describe the type of the message. There are two types of message: unicast ($T=0$) and multicast ($T=1$). The specific addresses of the source node and the destination node(s) are placed in the last field of the header in a row and the content of the message is located in the rest of flits (Payload).

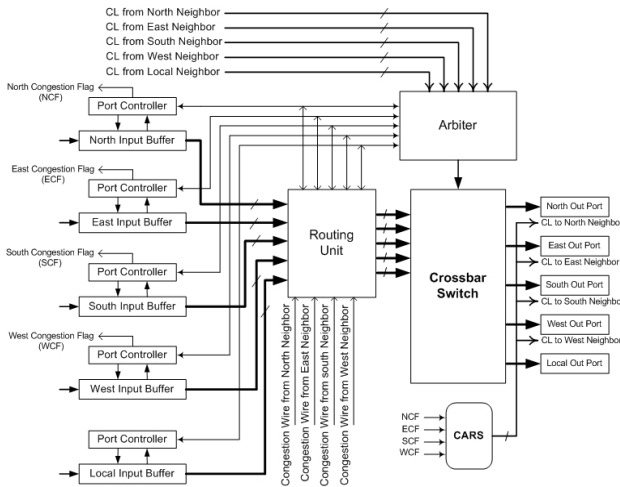


Fig. 3. The proposed routing structure

Router Structure: As shown in Fig. 3, each input port has a controller for handshaking and an input buffer. After receiving the flit header, first the routing unit determines to

which output port this packet should be sent, and then the arbiter requests for a grant to inject the packet to the proper output using the crossbar switch. It also controls the buffer status including empty and full states. In addition, the controller detects the sign of the rate at which the buffer is becoming occupied. 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). Each input port has a CF signal which informs its adjacent router about its congestion condition so that the congested input port should not be selected by the upstream router until the congestion condition is over. The router has a crossbar which establishes a connection path from an input port to an output port.

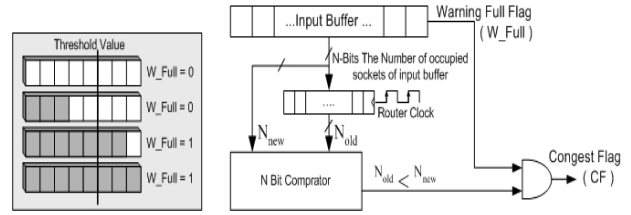


Fig. 4. Congestion detection circuit

For each output port the router uses an arbiter for selecting among simultaneous input requests to access the same output port. In order to detect whether the buffer status is critical or not, the entrance and departure rates of the buffer should be measured. For this purpose, the circuit shown in Fig. 4 is used. N_{new} is the number of occupied slots of the input buffer in the current cycle of the router clock and N_{old} is the same number but in the previous cycle of the 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 $N_{\text{new}} > N_{\text{old}}$ ($N_{\text{old}} > N_{\text{new}}$), it shows that the buffer is becoming full (empty). The status signal of the buffer 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 suggests that the congestion condition is traced using the signal W_{Full} which indicates the filling of the buffer. As shown in Fig. 4, CF will switch to high when both the W_{Full} signal and the positive rate for occupying the input buffer slots are detected. The Congestion Level (CL) of each router is computed by a module called Contention Aware Routing Selection (CARS). The CL is a binary number between 0 and 4 which is the sum of four CF's for four input ports (see Fig. 3 and Fig. 5). The CL for each router indicates its load level. For example, if the north and east input buffers of the router are congested ($NCF = 1$ and $ECF = 1$), then the CL value of the router will be 2. As illustrated in Fig. 5, the output of the CARS module of the router is sent to the corresponding input channels of its adjacent routers (downstream routers).

Output Selection: In the output selection, the router employs an address decoder which adopts the HAMUM routing algorithm to determine the proper output port. In HAMUM 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 or fallen, the message will be routed to p1.

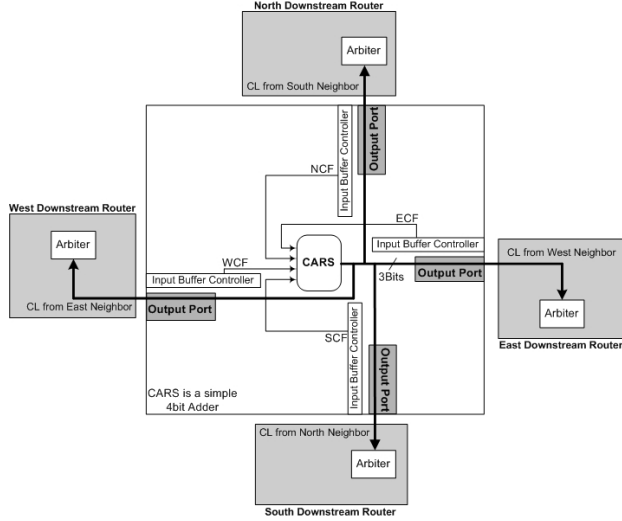


Fig. 5. Congestion Level Computation and Transmission Scheme

On the other hand, if the header type is a multicast message, the routing unit fetches the destination address from the header. 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.

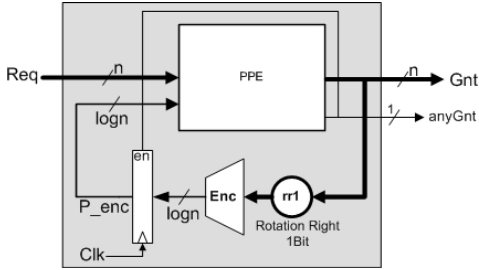


Fig. 6. Block diagram of a round-robin arbiter.

Input Selection: The proposed arbiter uses the WRR scheme derived from the RR policy. The scheme allows a weight to be assigned to each input port. The weight which specifies the number of packets to be transmitted when the router services that input port is proportional to the CL of the upstream router. This will assign different weights to the input channels of the routers for accessing the output channels through the arbitration process. The arbiter provides services for each input channel in turn in the round robin order. If the input channel buffer is empty, it will be skipped without being serviced. Fig. 6 shows a block diagram of a round robin arbiter [18]. The arbiter uses a Programmable Priority Encoder (PPE) unit to choose one highest priority request from n incoming requests (Req bus). In every arbitration cycle, PPE, which takes n 1-bit-wide requests and the $\log n$ -bit-wide pointer (P_enc) pointing to the current highest-priority request as its

inputs, chooses the first nonzero request value beyond (and including) $\text{Req}[P_enc]$. The output of the PPE is an n -bit-wide Gnt (grant) which has at most one nonzero bit and a 1-bit wide anyGnt signal which indicates if there has been at least one request. For updating the pointer, Gnt is loaded and rotated right one bit in rr1 unit (rotate right 1-bit register) whose output is encoded using the Enc unit and then latched for storing the next P_enc. Fig. 7 shows a block diagram of the Weighted Round Robin arbiter derived from the Round Robin scheme. The main difference between the two schemes is that WRR provides service to the input port in based on its CL. There are five registers four of which contain the CL of their upstream routers and one register is for the local router. The registers have three inputs and one output. If the register enable (En) is set, then the new CL value, which shows the CL of the upstream router, will be loaded in the register. After loading, the register operates as the down-counter for the service provided for this input port. While the zero signal (Zero) is not set (i.e., the register value has not reached zero) the register value will be decremented in each packet transmission cycle. When the register value reaches zero or the register enable (En) is reset, then the zero signal (Zero) will be set and subsequently the Enable of the rr1 unit is activated starting the update process for P_enc as was performed for the Round Robin scheme. In the situations where there are multiple input requests to the same output channel, each output channel arbiter will service the incoming requests according to their CL (weight). This mechanism resolves any possible starvation that might occur in arbiters based on priority scheme such as in CAIS.

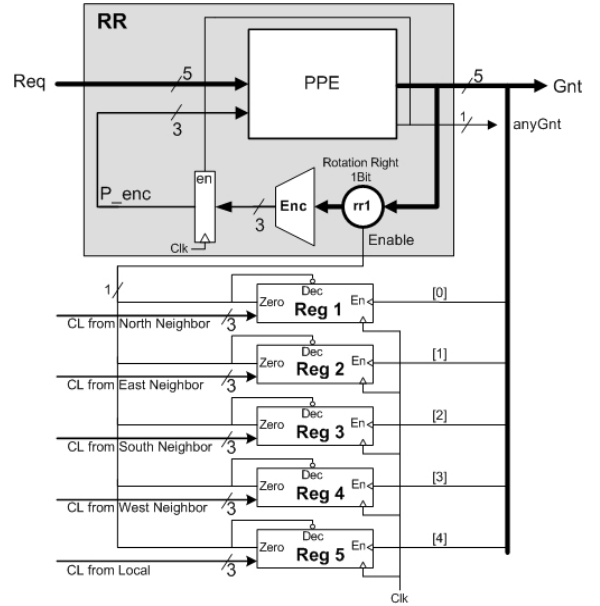


Fig. 7. Block diagram of a weighted round robin arbiter.

V. RESULTS AND DISCUSSION

Four different routing models, based on input-output selection, have been implemented to evaluate the proposed model (WRR-HAMUM). These models are: CAIS-OE (input selection is CAIS and output selection is Odd-Even), CAIS-MP (input selection is CAIS and output selection is Multi-Path), RR-OE (input selection is RR and output selection is

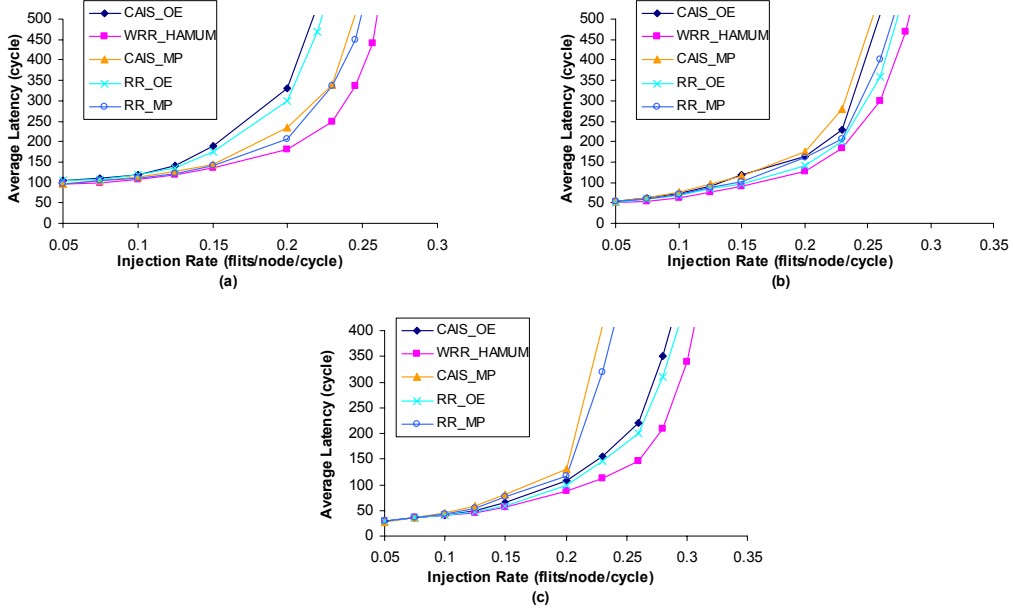


Fig. 8. Performance results with different loads in 8x8 2D-mesh under unicast, multicast, and mixed traffic profiles.

Odd-Even), RR-MP (input selection is RR and output selection is Multi-Path). An event driven NoC simulator in C++ which can calculate the average delay and the power consumption for the flit transmission has been developed. A two dimensional mesh configuration has been used 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 profile patterns. To calculate the power consumption, we have used Orion library functions [19]. For all switches, the data width is set to 32 bits, and each input channel has a buffer (FIFO) size of 10 flits with the congestion threshold set at 60% of the total buffer capacity. The packet size was assumed to be 5 flits. The time needed to generate the multicast messages is not considered, because we assumed the multicast messages are generated in the processing elements (PE). The array size has been considered 8x8.

A. Performance Evaluation

1) Multicast Traffic Profile

The first set of simulations were performed for a random traffic profile pattern. In this simulation, the PE generates five-flit messages and injects them into the network using the time intervals which are obtained based on the exponential distribution. In the multicast traffic profile, each PE sends a message to a set of destinations. A uniform distribution is used to construct the destination set of each multicast message [10]. The number of destinations has been set to 20. The average communication delay as a function of the average flit injection rate has been shown in Fig. 8(a). As observed from the results, the proposed mechanism leads to lower delay particularly, in high traffic loads. As described before and can be seen from Fig. 8(a), odd-even is not an efficient routing model for multicast traffics.

2) Unicast and Multicast (Mixed) Traffic Profile

In this set of simulation, we have employed a mixture of unicast and multicast traffic, where 70% of injected messages

are unicast messages and the remaining 30% 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 [10]. The unicast messages are also routed using HAMUM. Uniform [3] has been taken into account for unicast traffic generation. In the uniform traffic profile, each PE sends a message to any other PE in equal probability. This is determined randomly using a uniform distribution. In Fig. 8(b) the average communication latency of different models under the uniform traffic model for the unicast traffic is shown. As depicted in this figure, for this traffic, the proposed model outperforms the other models.

3) Unicast Traffic Profile

For appraising the unicast efficiency of WRR-HAMUM, The uniform traffic profile, where 100% of injected messages are unicast messages has been considered. Fig. 8(c) shows the simulation results for the uniform traffic. As depicted, when the injection rate is increased, WRR-HAMUM is superior to all of the other schemes. In brief, as the injection rate increases, the proposed algorithm leads to smaller average delays. This is due to the fact that the input selection uses WRR scheme which allows packet flows coming from congested paths to be serviced more often according to their congestion level. In contrast, in a RR scheme no matter how congested a path is, all packet flows are serviced equally. In the technique based on CAIS, congested input channels which have higher numbers of request are serviced more while the input channels with lower traffics may not be serviced leading to the starvation problem.

B. Power Dissipation

Using the simulator, the power dissipation of all models were calculated and compared under the unicast and multicast (mixed) traffic. The results for the average and the maximum power under mixed traffic are shown in Fig. 9(a) and (b) respectively. Both average and maximum power values are

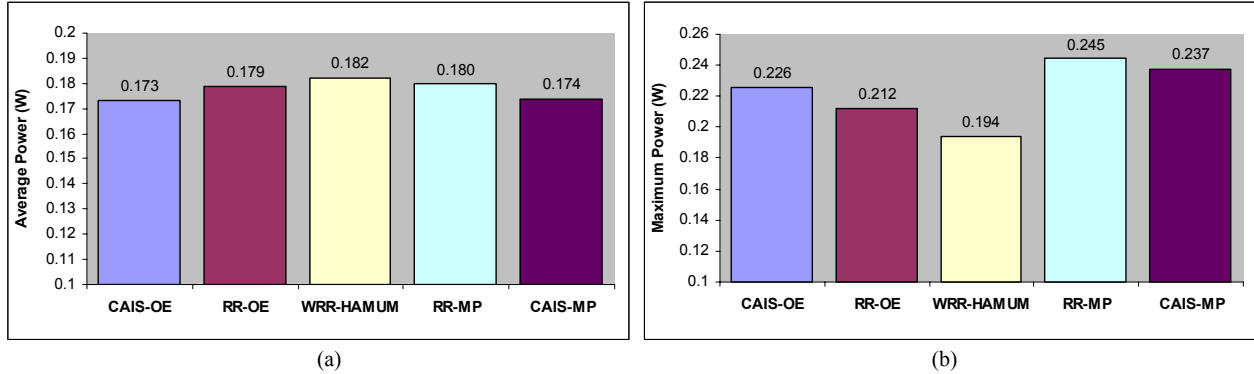


Fig. 9. (a) Average and (b) Maximum power dissipation results in 8×8 2D-mesh under mixed traffic profile.

computed near the saturation point, 0.23 (flits/cycle), under mixed traffic. We can notice that the peak power, compared to other schemes, is considerably lowered in our proposed scheme. This is achieved by smoothly distributing the power consumption over the network using the output selection scheme which reduces the number of the hotspots and, hence, lowering the peak power.

C. Hardware Overhead

To evaluate the area overhead of the presented model, and show the performance/area trade-off, aforementioned routers have been implemented with four different input-output selection schemes. The routers were described in VHDL and synthesized with Leonardo-Spectrum ASIC using the 0.09 μ m standard cell library. For all routers, the data width was set to 32 bits, and each input channel has a buffer size of 10 flits. The FIFOs were implemented in our design using registers in order to achieve better performance/power efficiency. Comparing the area cost of proposed model with RR-OE, RR-MP, CAIS-OE, and CAIS-MP introduces 1.3%, 1.5%, 2.4% and 3% additional overhead respectively.

VI. SUMMARY AND CONCLUSION

In this paper a new on-chip router architecture is proposed. The output selection of the presented router utilizes an adaptive routing algorithm supporting both unicast and multicast traffic while the input selection part of the router uses the weighted round robin arbitration. The adaptive output selection algorithm uses congestion flags to route packets through non-congested paths and consequently helps balance the traffic, whereas the WRR input selection assist in relieving nodes where congestion is formed. A C++ simulator was used to evaluate the efficiency of the proposed router. Under the multicast, unicast, and mixed traffic models and in high flit injection rates, the proposed model has the lowest average communication delay in comparison with the other models. It also reduces the average and maximum power dissipation of the network compared to other models under mixed traffic model.

ACKNOWLEDGMENT

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

REFERENCES

- [1] Luca Benini, Giovanni De Micheli, "Networks on Chips: A New SoC Paradigm," IEEE Computer, Vol. 35, No. 1, pp. 70-78, January 2002.
- [2] J. Duato, C. Yalamanchili, L. Ni, "Interconnection networks: an engineering approach", Morgan Kaufmann Publishers, 2003.
- [3] G. Chiu, "The Odd-Even Turn Model for Adaptive Routing," IEEE Tran. On Parallel and Distributed System, pp 729-738, July 2000.
- [4] J. Hu and R. Marculescu, "DyAD-Smart Routing for Networks-on-Chip," DAC 2004, pp: 260 - 263, 2004, San Diego, California, USA.
- [5] M. Ebrahimi, M. Daneshtalab, P. Liljeberg, H. Tenhunen, "HAMUM – A Novel Routing Protocol for Unicast and Multicast Traffic in MPSoCs," in Proceedings of 18th IEEE Euromicro Conference on Parallel, Distributed and Network-Based Computing (PDP), pp. 525-532, February 2010, Italy.
- [6] C.J. Glass and L.M. Ni, "The Turn Model for Adaptive Routing," Proc. 19th Ann. Int'l Symp. Computer Architecture, pp. 278±287, May 1992.
- [7] 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
- [8] M. Daneshtalab, M. Ebrahimi, S. Mohammadi, A. Afzali-Kusha, "Low distance path-based multicast algorithm in NOCs," in IET Computers and Digital Techniques, Special issue on NoC, Vol. 3, Issue 5, pp. 430-442, Sep 2009.
- [9] C. J. Glass and L. M. Ni, "The Turn Model for Adaptive Routing," Proc, Symp, Computer Architecture, pp. 278-287, May 1992.
- [10] 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.
- [11] P. Abad, V. Puente and J. A. Gregorio, "MRR: Enabling Fully Adaptive Multicast Routing for CMP Interconnection Networks," High Performance Computer Architecture (HPCA), 2009.
- [12] C. A. Zeferino, M. E. Kreutz, and A. A. Susin, "RASoC: A router soft-core for Networks-on-Chip," Designers Forum - DATE, pp. 198-203, France, 2004.
- [13] D. Wu, B. M. Al-Hashimi, and M. T. Schmitz, "Improving Routing Efficiency for Network-on-Chip through Contention-Aware Input Selection," In Proc. of 11th ASP-DAC, pp. 36 – 41, 2006.
- [14] E. Nilsson, M. Millberg, J. Oberg, and A. Jantsch, "Load distribution with the proximity congestion awareness in a network on chip," DATE, pp. 1126-7, Germany, 2003.
- [15] A. Demers, S. Keshav and S. Shenkar, "Analysis and Simulation of a Fair Queuing Algorithms," Proceedings of SIGCOMM '89, pp. 3-12, August 1989.
- [16] T. T. Ye, L. Benini, and G. De Micheli, "Packetization and routing analysis of on-chip multiprocessor networks," Journal of Systems Architecture, vol. 50, pp. 81-104, 2004.
- [17] J. Liang, et al., "aSOC: a scalable, single-chip communication architectures," in IEEE Int. Conf. on Parallel Architectures and Compilation Techniques, pp 37-46, Oct. 2000.
- [18] P. Gupta, N. McKeown. "Designing and Implementing a Fast Crossbar Scheduler," IEEE Computer Society Press, pp 20-28, Jan. 1999.
- [19] Wang, X. Zhu, L. Peh, S. Malik, "Orion: A Power-Performance Simulator for Interconnection Network," In Proc. Hot Interconnection, Stanford, CA, pp 294 – 305, August 2002.