Dual Congestion Awareness Scheme in On-Chip Networks

Masoumeh Ebrahimi, Masoud Daneshtalab, Juha Plosila, Hannu Tenhunen
Department of Information Technology, University of Turku, Finland
{masebr, masdan, juplos, hanten}@utu.fi

***Abstract*— One of the main factors limiting the performance of Networks-on-Chip is congestion. Routing algorithms perform an important role in distributing the traffic load over the network by providing alternative routing paths. The major parts of research works avoid congestion by considering the traffic condition in the forward paths and delivering packets through less congested paths. However, they do not consider the impact of the router arbitration in traffic distribution. On the other hand, traditional methods usually consider the traffic at the node level rather than a region level. In this paper, we propose a method named Dual Congestion Awareness (DuCA) which involves the congestion information about forward paths in the routing decision and backward paths in the router arbitration. DuCA can efficiently use the traffic information to transmit packets through less congested areas (using congestion information about forward paths) and steer traffic from highly congested regions to low congested areas (using congestion information in backward paths). Finally, DuCA is a region-based approach, providing a wider view of the network traffic.**

I. INTRODUCTION

Networks-on-Chip is a promising solution for the scalability and the long interconnect problems of bus-based approaches [1][2][3]. The performance and efficiency of NoCs largely depend on the input selection [4][5][6][7] and output selection [8][9][10][11] methods exploited by on-chip routers.

The input selection method chooses one of input channels to get access to the output channel, done by an arbitration process. The arbiter could follow either non-priority or priority scheme. In the priority method, when there are multiple input port requests for the same available output port, the arbiter grants access to the input port having the highest priority level. The priority scheme can help to flatten the network congestion by giving higher priority level to traffic coming from congested areas [6][12].

The output selection method, using a routing algorithm, determines which output channel should be chosen for a packet arrived from an input channel. The output selection function can be classified as either congestion-oblivious [13][14] or congestion-aware [15][16][17] schemes. In congestion-oblivious algorithms, such as dimension-order routing, decisions are independent of the congestion condition of the network. This policy may disrupt the load balance since the network status is not considered. In congestion-aware algorithms, the congestion status of a number of nodes is used in the routing decision. A group of congestion-aware algorithms considers local traffic condition in which each router analyses the congestion condition of itself and adjacent routers to choose the output channel. Routing decisions based on local congestion information may lead to an unbalanced distribution of traffic load. These algorithms are efficient when the traffic is mostly local, i.e. cores communicate with other ones close to them [18], but they are unable to solve the global load balance problem via making local decisions [19].

In this paper, we propose an efficient congestion-aware routing method for multiprocessor architectures, named Dual Congestion Awareness (DuCA) method. DuCA can significantly improve the performance of NoCs by 1- employing the global congestion information about backward paths (upstream routers) in the arbitration unit of routers. 2- delivering packets through a less congested route using the global congestion information about forward paths (downstream routers).

The paper is organized as follows. In Section III, the basic idea of the proposed method is introduced. In Section III, recent congestion-aware proposals are discussed. In Section IV, the proposed DuCA method is explained. The results are reported in Section V while the summary and conclusion are given in the last section.

II. THE BASIC IDEA BEHIND DUAL CONGESTION-AWARENESS TECHNIQUE

In network-based multiprocessor architectures, network congestion affects the system performance considerably [19][20][21][22]. Several methods have been presented to address the network congestion problem. These methods mainly focus on routing packets through less congested paths and avoiding additional traffic to the congested area. However, they cannot utilize the congestion information in the scheduling process of routers. The DuCA method utilizes the global congestion information of upstream routers (backward paths) in the arbitration process of routers and downstream routers (forward paths) in the routing decision. The main ideas of DuCA consist of two parts: input selection function and output selection function.

A. Input Selection Function

The input selection function of DuCA uses a priority-based approach tries to prioritize packets coming from congested areas. In order to explain the DuCA method, let us consider the example in Fig. 1(a) where a 4×4 mesh network with three congested nodes 5, 6 and 10 is illustrated. If a fair arbitration mechanism is used at the router 9, arriving packets from the congested routers 5 and 10 will have the same chance to win the arbitration as the packets from the routers 8 and 13. Accordingly, the router 9 can be a bottleneck for the packets which are leaving the congested area. This bottleneck problem can be resolved by giving higher priority to the arrived packets from the routers 5 and 10 to access the output ports at the

router 9. In contrast, packets arriving from the routers 8 and 13 should wait in the input buffers of the router 9 before accessing the output channel which increases the congestion condition at the router 9 slightly. In other words, the traffic of highly congested areas is distributed over low congested ones. Fig. 1 (b) depicts the spread of traffic congestion over the network where packets arriving from congested nodes (i.e. nodes 5, 6 and 10) get more chance to win the arbitration among the neighboring nodes (i.e. nodes 1, 2, 7, 11, 14, 9 and 4), and similarly, the priority-based arbitration is performed in the rest of the routers. In sum, the idea behind the input selection function of the DuCA method is to allow congested nodes to forward their buffered packets rapidly.

B. Output Selection Function

The output selection function of DuCA attempts to send a packet through a direction which is less congested. Consider the case, for example, when the source and destination are at nodes 0 and 15 and the packet is already at node 5 (Fig. 1(a)). The packet can be delivered either through the node 6 or node 9. Since the node 9 is less congested, the packet should be sent through this node, thus alleviating the traffic load in the congested nodes.

In Section IV it would be explained that both input selection and output selection functions of DuCA are made based on the region-based congestion information rather than node-based scheme which is discussed in this example.

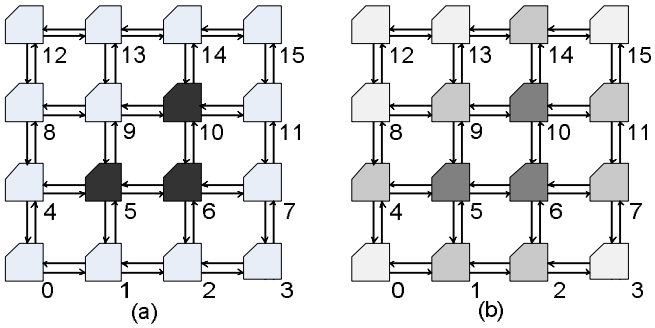

Fig. 1. An example of traffic distribution using the DuCA method

III. RELATED WORK

Several methods are presented in the realm of NoCs in order to balance the traffic load over the network. Almost all of these algorithms consider the traffic condition in forward paths and a few of them take backward paths into consideration. Here, we explain and discuss some of the recent works in this area.

A. ANoC: Congestion-Aware Selection Method in Agent-based Network-on-Chip

In the Agent-based Network-on-Chip (ANoC) structure [15], the network is divided into several clusters in which a cluster includes four routers and a cluster agent. The design consists of two separate mesh networks: main data network and lightweight congestion network. The main data network connects the routers to each other to propagate packets over the network; while in the congestion network, cluster agents are communicated with each other to spread the congestion information. Each cluster agent performs two simple tasks. First, it collects the congestion information from the attached routers (local routers) and distributes the information to the neighboring cluster agents as well as the local routers; second, it receives the congestion information from the adjacent cluster agents and transmits them to the local routers. By distributing congestion information over the network, routing decision can be assisted by the local and non-local congestion information received from different regions of the network.

The main advantage of this method is in providing a wider view of the network congestion by considering a group of nodes (regions) rather than single nodes. The weakness of this method is that it only uses the congestion condition of forward paths in the routing decision.

B. CATRA: Congestion Aware Trapezoid-based Routing Algorithm

The idea of the ANoC approach is improved in the CATRA [16] routing algorithm. CATRA tries to collect and utilize the congestion information for just enough number of nodes in the network. In CATRA, the passing probability of packets through the nodes is calculated. Based on this measurement, the nodes with high passing probabilities form a region of trapezoid shapes. The local and non-local information is gathered from the nodes in these regions (regarding the current node) that not only are more likely to be chosen as intermediate nodes in the routing path but also provide up-to-date information for a given node.

One of the advantages of CATRA is in efficiently choosing the groups of nodes which is determined based on the passing probability of packets through the nodes. Similar to ANoC approach, it only involves the congestion status of forward paths in the routing decision.

C. DuQAR: Dual Q-routing Adaptive Learning Rate

DuQAR [17] introduces a new idea based on Q-learning approaches in NoCs. In this method, the network condition is learned in run time and then utilized in the routing decision. Each router maintains a Q-table to store estimated latencies from the source to each destination nodes in the network. As packets move in the network, Q-tables incorporate more global information. Thereby, the contents of Q-tables efficiently represent the traffic conditions in the network. In the DuQAR method, the learning rate dynamically changes according to the congestion condition of the network. For this purpose, a congestion detection method is presented to measure the average of free buffer slots in a specific time interval. This value is compared with maximum and minimum threshold values and based on the comparison result, the learning rate is updated. If the learning rate is a large value, it means that the network gets congested and global information should be more emphasized than local information. In contrast, local information is more important than global information when a router receives a few packets in a time interval.

DuQAR utilizes of the congestion condition of both forward and backward paths. However, the main disadvantage of DuQAR is in the large area overhead due to using Q-tables. This approach is not scalable and the area overhead increases significantly with larger network sizes.

D. HARAQ: Highly Adaptive Routing Algorithm using Q-Learning

The idea of HARAQ [23] is to take the advantages of both learning methods and non-minimal paths. HARAQ claims that minimal methods can propagate messages over at most two directions at each node, limiting the performance of NoCs. When the shortest paths are congested, sending more messages through them can deteriorate the congestion condition considerably. In HARAQ, an adaptive routing algorithm for on-chip networks is presented that provides a wide range of alternative paths (either minimal or non-minimal) between each pair of source and destination nodes. The selection function plays an important role in the HARAQ method as selecting longer paths with higher congestion has a significant negative impact on latency. To address this problem, an optimized and scalable learning method is utilized. The learning method is based on local and global congestion information and can estimate the latency from each output channel to the destination region. This approach imposes a lower area overhead than DuQAR as the entries of a Q-table are reduced from the number of destinations into the number of destination regions in the network.

Unlike DuQAR, HARAQ is a scalable approach. In addition, it benefits from an efficient distribution of traffic over the network by combining non-minimal paths and learning approaches. However, this approach has less accurate congestion information in Q-tables compared with DuQAR. The reason is that the latency estimation is obtained from each source to each destination region rather than each pair of source and destination nodes in DuQAR.

E. GLB: Global Load Balancing Method

In GLB [24], packets aggregate and carry congestion information along a path they route. Consequently, this information contains a global view of the path from where a packet is routed. This global congestion information is used as a metric in the arbitration unit in order to give more priority to packets arriving from congested area. This method makes more attention on the congestion information on backward paths with less attention to forward paths.

In sum, ANoC, and CATRA are region-based methodologies, considering only forward paths. DuQAR and HARAQ are learning approaches. DuQAR has a large area overhead while HARAQ only makes attention on forward paths. Finally, GLB considers both forward and backward paths with highly more attention on backward paths. GLB is not a region-based approach, providing a limited view of congestion. The main characteristics of our proposed approach, DuCA, are as follows: 1- it is a region-based approach 2- it is a scalable approach without using tables 3- it considers the congestion information of both forward and backward paths in the routing decision 4- it has no additional overhead over region-based ANoC approach.

IV. The Presented Approach: Dual Congestion Awareness Method (DuCA)

A. Congestion Information Propagation Mechanism

The network architecture of the DuCA method is shown in Fig. 2 which is similar to the ANoC approach discussed in section III-c.

To compute the Congestion Level (CL) of each router, the buffer status of each input buffer is used by the cluster agent. To show the buffer status, we use a signal named Congestion Status (CS). The CS signal of every input buffer is determined by a history-based scheme which captures the input buffer threshold signal. A 4-bit shift register is adopted to store the threshold signal whenever a new flit enters or leaves the buffer (flit events: flit_tx or flit_rx). That is, in each flit event, if the number of occupied buffer cells is larger (smaller) than a threshold value, the threshold signal is assigned to one (zero) and stored in the shift register. A majority function is used to determine the CS value of an input port. The CL value of each router is computed by summing up the CS signals received from the input ports. Afterward, each cluster agent receives and combines the CL values of local routers in order to transfer the CL string to the neighboring cluster agents as well as the local routers.

B. Input Selection Function

By the propagation mechanism, each router is aware of the congestion status in four neighboring clusters as well as its own cluster. It can be used in routers to prioritize packets. The input selection function examines the priority values of competing packets and gives the priority to packets coming from congested clusters. In order to prevent starvation, each time after finding the highest value, the priorities of defeated packets are incremented. If the source cluster is the same as the current cluster, the congestion value of the current cluster is used in the routing process; otherwise the congestion level of the upstream cluster takes into consideration.

C. Output Selection Function

The routing function returns a set of admissible output channels that can be used for delivering packets. We utilize two virtual channels along the X and Y dimensions to provide fully adaptiveness. In DuCA, the output selection function chooses one of them based on the congestion condition of the neighboring clusters.

If the destination is located in the same row or column as the neighboring cluster, only the number of occupied buffer cells at the corresponding input buffers of the neighboring nodes is considered as the congestion metric. Otherwise, the congestion levels of two neighboring clusters are compared with each other.

In Fig. 2, let us assume that a packet is sent from the source node 0 to the destination node 35 where the source and destination nodes are located at cluster 0 and cluster 8,

respectively. Since the destination node is located in a non-neighboring cluster, the congestion levels at neighboring clusters 1 and 3 are compared with each other (the situation is the same for all the nodes within the cluster 0). After passing the intermediate nodes in the cluster 0, the packet is sent to the cluster 3 which is less congested than the cluster 1. At cluster 3, the packets coming from the cluster 4, cluster 0, and cluster 3 may compete against each other for receiving resources in order to be routed toward the cluster 6. Among them, packets coming from the cluster 4 have the highest priority as it is highly congested. By increasing the priority of defeated packets, finally the packets coming from the cluster 0 win the arbitration in the routers of cluster 3 and are delivered to the cluster 6. At cluster 6, there are two types of packets competing for being sent to the cluster 7: those originated at cluster 6 and those receiving from the cluster 3. Since the cluster 6 is more congested, its packets receive higher priority in the arbitration process. As been already mentioned, when the destination cluster is located in the same row as the current cluster, the number of occupied buffer slots is considered as the congestion metric rather than the congestion level of neighboring clusters. Therefore, the number of occupied buffer slots is used in the routing decision from this node onwards. Similar input selection function is applied for the remaining path.

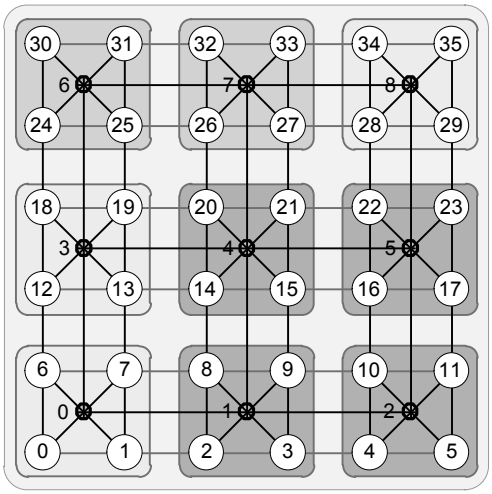

Fig. 2. The network architecture in the DuCA approach

V. EXPERIMENTAL RESULTS

To evaluate the proposed DuCA method, a NoC simulator is implemented with VHDL. The simulator models all major components of the NoC such as network interfaces, routers, and wires [25]. We use ANoC (section III-C) and GLB (section III-D) approaches as our baseline models.

A. System Configuration

As shown in Fig. 3, a 2D mesh platform is configured where eighteen nodes are assumed to be processors and other eighteen nodes are memories. The processors are 32b-AXI and the memories are DRAM (t_{RP}-t_{RCD}-t_{CL}=2-2-2, 32b). We adopt a commercial memory controller with the memory interface, DDR2SPA module from Gaisler IP-cores [27] where it is placed between the memory and the slave network interface. The link width, buffer depth, and traffic type are the other parameters which are specified for the simulator. The routers utilize wormhole switching. For all routers, the data width (flit size) is set to 32 bits and the buffer depth of each channel is set to 5 flits. For the request, the command and all its control bits are stored in the first flit of the packet, the memory address is set in the second flit, and the write data are appended at the end. For the response message, the control bits are included in the first flit while the read data are appended at the end if the response relates to a read request. As the performance metric, we use latency defined as the number of cycles between the initiation of a request operation issued by a master (processor) and the time when the response is completely delivered to the master from the slave (memory). The request rate is defined as the ratio of the successful read/write request injections into the network over the total number of injection attempts. All memories (slave cores) can be accessed simultaneously by each master core continuously generating memory requests.

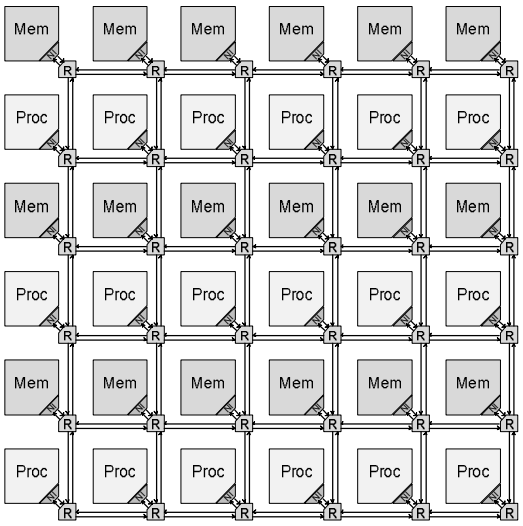

Fig. 3. 6x6 NoC layout.

B. Performance Evaluation

We consider uniform and non-uniform/localized synthetic traffic patterns for the evaluation and we expect applications stand between these two synthetic traffic patterns [28]. The random traffic represents the most generic case, where each processor sends read/write requests to memories with a uniform probability. Hence, the target memory and request type (read or write) are selected randomly. Eight burst sizes, from 1 to 8, are stochastically chosen according to the data length of the request. In the non-uniform mode, 70% of the traffic is local requests, where the destination memory is one hop away from the master core, and the remaining 30% of the traffic is uniformly distributed to the non-local memory modules.

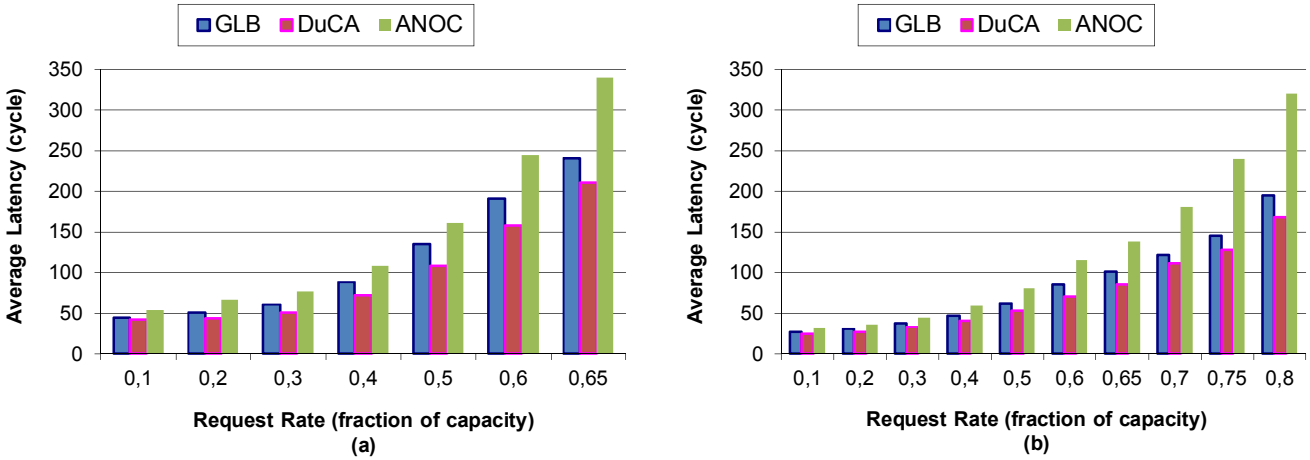

Fig. 4. Performance evaluation under (a) the uniform and (b) non-uniform traffic models.

Fig. 4 (a) and (b) show the simulation results under the uniform and non-uniform traffic models, respectively. In the presented configuration, the on-chip network utilizing the proposed DuCA method is compared with the networks equipped with GLB and ANoC methods. As demonstrated in both figures, compared with GLB and ANoC architectures, the NoC using DuCA reduces the average latency when the request rate increases under the uniform and non-uniform traffic models. The performance gain near the saturation point (0.6) under the uniform and non-uniform traffic models is about 17%~35% and 16%~38%, respectively.

C. *Physical Implementaiton*

A router of each platform, DuCA, ANoC, and GLB, is synthesized with Synopsys Design Compiler using the 65nm standard CMOS technology with a timing constraint of 1 GHz for the system clock and supply voltage of 1 V. The synthesized netlist is then again verified through post synthesis simulations. Finally, we perform place-and-route, using Cadence SoC Encounter, to have precise power and area estimation in wire-dominated structures. The power dissipation of each scheme is calculated under the uniform traffic model near the saturation point (in the aforementioned 6×6 2D mesh) using Synopsys PrimePower. The layout areas and power consumptions of presented schemes are listed in Table 1. As the results, reported after place-and-route, the GLB router has lower area and power consumption in comparison with the ANoC and DuCA routers.

Table 1. Hardware costs.

Router	Area (mm^2)	Power (mW)
GLB Router	0.1873	59
ANoC Router	0.1913	66
DuCA Router	0.1954	69

VI. SUMMARY AND CONCLUSION

Network congestion can limit the performance of NoC due to increased transmission latency. Most of all recent congestion-aware techniques target to alleviate congestion in the network by delivering packets to the less congested paths. In other words, almost none of them take the information about backward paths into consideration. DuCA tries to utilize the congestion status of both forward and backward paths in the routing decision. Moreover, DuCA provides a better view of the network congestion by considering traffic at the region-level rather than node-level. Experimental results demonstrate that the DuCA method provides significant improvement in average network latency compared with traditional methods.

REFERENCES

[1] J. Duato, et al., "Interconnection Networks: An Engineering Approach." Morgan Kaufmann, 2002.
[2] L.Benini and G.De Micheli, "Networks on chips: a new SoC paradigm", IEEE Computer, January 2002.
[3] MH Neishaburi and Z Zilic, "A Fault Tolerant Hierarchical Network on Chip Router Architecture" In proc. of Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), 2011.
[4] C. A. Zeferino, et al., "RASoC: A Router Soft-Core for Networks-on-Chip", Proceedings of DATE'04, pp. 1530-1591, 2004.
[5] M. Daneshtalab, et al., "Adaptive Input-output Selection Based On-Chip Router Architecture," Journal of Low Power Electronics *(JOLPE)*, Vol. 8, No. 1, pp. 11-29, 2012.
[6] D. Wu, et al., "Improving Routing Efficiency for Network-on-Chip through Contention-Aware Input Selection," In Proc. of 11th ASP-DAC, pp. 36 – 41, 2006.
[7] M. Daneshtalab, et al., "Distributing Congestions in NoCs through a Dynamic Routing Algorithm based on Input and Output Selections," in Proceedings of 20th IEEE Conference on VLSI Design (VLSID), pp. 546-550, Jan 2007, India.
[8] P. Lotfi-Kamran, et al., "BARP-A Dynamic Routing Protocol for Balanced Distribution of Traffic in NoCs", In DATE conference, pp. 1408-1413, 2008.
[9] W. Feng and K. G. Shin, "Impact of Selection Functions on Routing Algorithm Performance in Multicomputer Networks", In International Conference on Supercomputing, pages 132–139, 1997.
[10] M. Daneshtalab, et al., "A Generic Adaptive path-based routing method for MPSoCs," Journal of Systems Architecture (JSA-elsevier), Vol. 57, No. 1, pp. 109-120, 2011.
[11] M. Dehyadegari, et al., "An Adaptive Fuzzy Logic-based Routing Algorithm for Networks-on-Chip," in Proceedings of 13th IEEE/NASA-ESA International Conference on Adaptive Hardware and Systems (AHS), pp. 208-214, June 2011, USA.
[12] M. Ebrahimi, et al., "An Efficent Dynamic Multicast Routing Protocol for Distributing Traffic in NOCs," in Proc. of 12th IEEE/ACM Design,

Automation, and Test in Europe (DATE), pp. 1064-1069, April 2009, France.
[13] H.G. Badr, S. Podar, "An optimal shortest-path routing policy for network computers with regular mesh-connected topologies", v.38, I.10, pp.1362-1371, 1989.
[14] A. Sobhani, et al., "Dynamic Routing Algorithm for Avoiding HotSpots in On-chip Networks," in Proceedings of 2th IEEE International Conference on Design & Technology of Integrated Systems in Nanoscale Era (DTIS), pp. 179-183, Sep 2006, Tunis.
[15] M. Ebrahimi, et al., "Agent-based On-Chip Network Using Efficient Selection Method," in Proceedings of 19th IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC), pp. 284-289, Oct 2011, Hongkong.
[16] M. Ebrahimi, et al., "CATRA-Congestion Aware Trapezoid-based Routing Algorithm for On-Chip Networks," in Proceedings of 15th ACM/IEEE Design, Automation, and Test in Europe (DATE), pp. 320-325, Mar. 2012, Germany.
[17] F. Farahnakian, et al., "Adaptive Reinforcement Learning Method for Networks-on-Chip," in Proceedings of 16th IEEE 12th International Conference on Embedded Computer Systems: Architectures, MOdeling, and Simulation (SAMOS XII), pp. , July 2012, Greece.
[18] L. P. Tedesco, et al., "Implementation and evaluation of a congestion aware routing algorithm for networks-on-chip" pp. 91-96, 2010.
[19] P. Gratz,, et al, "Regional Congestion Awareness for Load Balance in Networks-on-Chip", Proc. HPCA, pp. 203–214, 2008.
[20] A. Singh, et al., "GOAL: A Load-Balanced Adaptive Routing Algorithm for Torus Networks", In International Symposium on Computer Architecture, pages 194–205, 2003.
[21] A. Singh et al, "Globally Adaptive Load-Balanced Routing on Tori" IEEE Computer Architecture Letters, v.3, I.1,pp.2-6, 2004.
[22] M. Ebrahimi, et al., "Exploring Partitioning Methods for 3D Networks-on-Chip Utilizing Adaptive Routing Model," in Proceedings of 5th ACM/IEEE International Symposium on Networks-on-Chip (NOCS), pp. 73-80, May 2011, USA.
[23] M. Ebrahimi, et al., "HARAQ: Congestion-Aware Learning Model for Highly Adaptive Routing Algorithm in On-Chip Networks," in Proceedings of 6th ACM/IEEE International Symposium on Networks-on-Chip (NOCS), pp. 19-26, May. 2012, Denmark.
[24] M. Daneshtalab et al., "GLB - Efficient Global Load Balancing Method for Moderating Congestion in On-Chip Networks," In proc. of 7th IEEE International Symposium on Reconfigurable Communication-centric Systems-on-Chip (ReCoSoC), pp. , July 2012, UK.
[25] M. Daneshtalab et al., "Memory-Efficient On-Chip Network with Adaptive Interfaces," IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems (IEEE-TCAD), Vol. 31, No. 1, pp. 146-159, Jan 2012.
[26] MH Neishaburi, Z Zilic, "ERAVC: Enhanced reliability aware NoC router," in prof. of 12th International Symposium on Quality Electronic Design (ISQED), pp. 1-6, 2011.
[27] Gaisler IP Cores, http://www.gaisler.com/products/grlib/, 2009.
[28] R. Das, et al., "Design and Evaluation of a Hierarchical On-Chip Interconnect for Next-Generation CMPs," in proc. of 15th International Symposium on High-Performance Computer Architecture (HPCA), pp. 175-186, 2009.