JACET

J. ADV COMP ENG TECHNOL, 4(2) Spring 2018

Congestion Estimation of Router Input Ports in Network-on-Chip for Efficient Virtual Allocation

Ali Ramezanzad[1]; Midia Reshadi[2]; Akram Reza[3]
1,2,3 Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran.
(reshadi@srbiau.ac.ir)

Received (2018-04-16) Accepted (2018-06-21)

Abstract: *Effective and congestion-aware routing is vital to the performance of network-on-chip. The efficient routing algorithm undoubtedly relies on the considered selection strategy. If the routing function returns a number of more than one permissible output ports, a selection function is exploited to choose the best output port to reduce packets latency. In this paper, we introduce a new selection strategy that can be used in any adaptive routing algorithm. The intended selection function is named Modified-Neighbor-on-Path, the purpose of that is handling the condition of hesitation happening when the routing function provides a set of acceptable output ports. In fact, number of inquiries that each router has sent to its neighbors in determined past cycles is a new parameter that can be combined with number of free slots of adjacent nodes in the latest selection function named Neighbor-on-Path. Performance analysis is performed by using exact simulation tools under different traffic scenarios. Outcomes show how the proposed selection function applied to West-first and North-last routing algorithms outperforms in average delay up to 20 percent on maximum and an acceptable improvement in total energy consumption.*

Keywords: *Network-on-chip(NoC), Modified-Neighbor-on-Path(MNoP), adaptive routing, selection strategy, performance analysis, congestion, contention, selection function, routing, Neighbor-on-Path (NoP).*

How to cite this article:
Ramezanzad A, Reshadi M, Reza A. Congestion Estimation of Router Input Ports in Network-on-Chip for Efficient Virtual Allocation. J. ADV COMP ENG TECHNOL, 4(2) Spring 2018 : 79-86

I. INTRODUCTION

Network-on-chip is introduced as a contemporary method for implementing reliable, scalable, flexible and modular capabilities in Multi-processor System-on-Chips. The scheme of on-chip networks has been represented in order to handle the scalability matters provided in shared-bus networks and facilitate to implement a huge number of processing cores on a limited die. Nevertheless, there are still many restrictions in this domain, such as providing economic energy consumption, minimizing delay, on a restricted die area. The common performance of a NoC is relevant to many network assets, like routing function, topology, flow control, deadlock-free and selection strategy [1].

The most imperative anxiety of the latest researches rely on the design of reliable, flexible, high-performance, low-cost, on-chip router architectures, the solutions for contention and congestion-aware selection function, and the development of deadlock-free selection schemes. Selection strategy undoubtedly affects the general performance of any adaptive routing algorithm [2] and is the main focus of this paper.

The switching technique in this paper is wormhole. In wormhole switching, each packet is divided into a order of flow control units (flits) [3]. In adaptive routing, if the number of output ports is identified by the router consists at least two non-reserved output ports, the router must choose one of them. The further step required to manage these conditions of hesitation depends on the functionality of selection functions [4]. Fig. 1 shows the condition of hesitation for a 2D mesh topology, Odd-Even [5] routing algorithm [6], [7] and 4 flits buffer size. As can be seen, on average, the percentage of conditions of hesitation is more than 36 percent. [8]

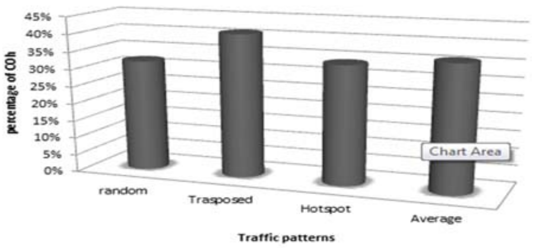


Fig.1. Percentage of conditions of hesitation for different traffic scenarios [8].

In this paper, we represent an improved output selection function named "Modified-Neighbor-on-Path", which designed with the intention of distributing traffic and avoiding congestion in network as much as possible in order to obtain better latency outcomes.

The reminder of the paper is arranged as follows: In section II an outline of the related work is introduced, tabling some of the selection functions provided lately. Section III explains the proposed selection strategy by describing the functionality of the function precisely. Section IV is devoted to experimental results that the proposed selection function is compared with NoP selection function in parameters such as latency and energy consumption. Ultimately, conclusions and future works are mentioned in section V.

II. RELATED WORKS

In order to describe the architecture of the proposed selection function, first of all, we should review a number of related works which could clarify the scheme of proposed selection function.

Authors of [8] have represented a new selection function for adaptive routing algorithms in on-chip networks, named Neighbor-on-Path. The scheme of Neighbor-on-Path related to the goal of choosing the best output port which tries to minimize the congestion in NoC as much as possible. It solves the condition of hesitation by identifying congestion states from non-adjacent nodes.

In [9] a new selection strategy that can be used with any adaptive routing algorithm has been represented, named Path-aware, which exploits turn model in order to avoid path-based contention. Authors of [10] have represented a power-aware selection strategy, opting power with performance, as factors for choosing the best output port. In [11] the possibility of implementing high-performance and low-cost routing algorithms is evaluated in order to balance network traffic. This is performed by the data is relevant to connections in applications, such as connection bandwidth and connection topology.

In [12] a fuzzy controller is used to merge two factors are relevant to congestion, i.e. set of crossbar requests and number of free buffer slots that can be used with adaptive routing algorithms. In [13] a technique for improving input selection in routers has been considered. Despite of the latest works that had concentrated on improving output selection functions, in this paper, a mechanism, named Contention-Aware Input Selection, is represented in order to improve routing performance with choosing the congestion-free input port based on the congestion status of the upstream switches that could in turn eliminate possible network congestion.

Authors of [14] have represented a new way for upgrading routing in on-chip networks that exploits a self-optimized mechanism with the goal of refraining hotspot nodes when sending packets. Authors of [15] have introduced a lightweight mechanism, named Regional Congestion Awareness, in order to enhance network balance when exploits adaptive routing algorithms. Presenters of [16] have introduced a new selection strategy, named Look-Ahead Traffic-aware Execution, which can be used for special applications. The major aim is distributing traffic load, aiming to better performance outcomes.

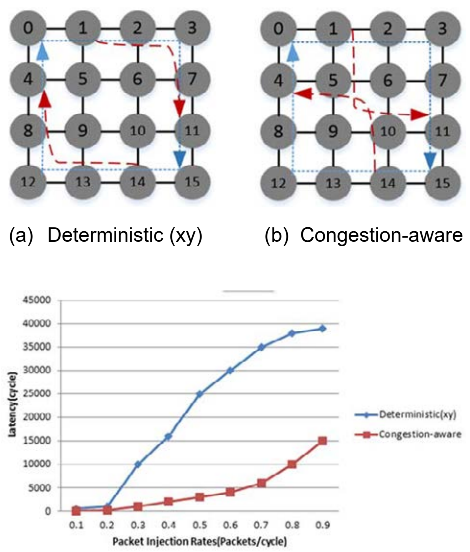
Authors of [17] have represented Destination-Based Adaptive Routing in order to introduce concepts for handling matters related to intra- and inter-application for output channel selection in consolidated workloads. In [18] an adaptive routing algorithm has been represented, which named Traffic- and Throttling- Awareness Routing in order to handle traffic congestion matters happened by throttling of transient-temperature control, achieving to distribute the network traffic.

In [19] a Regional ACO-based regional routing algorithm has been represented, which takes advantage of static and dynamic regional table organization mechanism, with the aim of achieving a method for more distributing load of the network.

III. MNoP SELECTION FUNCTION

A. Motivational Example

One of the most significant concerns in this paper is the selection policy refrain from congested zones as much as possible. As pointed out before, selection function is one of the major items that influence the performance in on-chip networks.



(a) Deterministic (xy) (b) Congestion-aware

(c) latency comparison

Fig. 2.Congestion-awareness effect on latency

As results show in Fig. 2, two different routing functions have been supposed: in deterministic (xy), despite of congestion-aware routing functions as depicted in Fig 2.a and Fig 2.b packets overlap each other. In fact, contention-aware routing function avoids situations guiding to contention as much as possible. Therefore, as Fig 2.c shows, it outperforms in metrics such as latency and total energy consumption. So, this paper also concentrate on this concept in order to represent a selection strategy which is contention-aware and try to choose the output port that is out of congestion.

B. MNoP Selection Function Description

In this section, the architecture of the proposed selection function is described precisely. The selection functions as routing algorithms, executes at every node. One of the reasons for implementing selection function is selecting the best output port. In the following paragraphs, a wormhole-based switching mechanism on a mesh topology is supposed.

We have chosen West-First and North-last as fixed routing algorithms, so the proposed selection function is applied to these routing algorithms. As pointed out in section I, in a routing algorithm, first the routing function provides a set of output ports as nominees, considering the routing limitations established by the routing algorithm. Next, it is the role of the selection function to choose the best output port, considering the network conditions, trying to avoid congestion in the network and minimize contention as much as possible. Also, in this section, the functionality of MNoP selection function, based on the number of free-slots and one additional parameter we named inquiries calculated from the adjacent nodes is explained.

This paper focus relies on the number of inquiries that each candidate node has sent in determined past cycles. The purpose of inquiry in this paper is as same as the number of requests each node has sent in previous time duration. As mentioned before, this paper main purpose is identifying the congested nodes as much as possible. So, this variable is a new parameter which lets to identify the congestion in router input ports.

In order to estimate the congestion in router input ports this parameter is combined with number of free slots to avoid congestion from source to destination. This combination will be based on an appropriate formula which is obtained based on practical simulation results.

In proposed selection strategy, our purpose of determined past cycles is two hop counts duration. It is assumed that for sending flits through two hop counts it is needed at least two clock cycles in order to avoid hardware overhead. Because if this variable goes over a specified amount it needs additional buffer space in order to hold this special variable. Finally, a counter in each node for counting number of inquiries that each node has sent to adjacent nodes is used. This counter will be reset every two clock cycles. It is supposed that two clock cycle is as two hop counts.

In the following example, the proposed method, based on the MNoP computation is demonstrated.

Let us first notice to some of the ideas behind the MNoP selection function. A detailed explanation of the algorithm executed at each node is explained in section B. For example, let us assume a mesh topology. The first flit of the packet entered into input port of node (1, 1) and adaptive routing algorithm selected nodes (2, 1) and (1, 2) as appropriate candidate nodes, assume that, packets are moving toward the node (3,3), which is a destination node, as depicted in Fig. 3a. Node (1, 1) must choose whether to transmit the header flit (and other body flits) to node (2, 1) or node (1, 2). As depicted in Fig. 3b, the solution is that node (1, 1) would choose a better option if only it had some information about the input buffer condition and number of inquiries of nodes that locates beyond nodes (2, 1) and (1, 2). In addition, we combine number of inquiries with number of free slots in the latest two clock cycles in order to our new selection function has better option for sending flits in as idle as possible way. Relying on the ultimate destination node defined in the header, not whole of the data introduced is finally beneficial to node (1, 1). In order to determine the next appropriate node for the packet, the routing function will ignore all buffer availability and number of inquiries situation which are not on a appropriate routing direction. As a result, an additional information that node (1, 1) requires is considering which nodes can really be attained by the header flit. So, we represent the idea of MNoP: Node (1, 1) exploits the routing

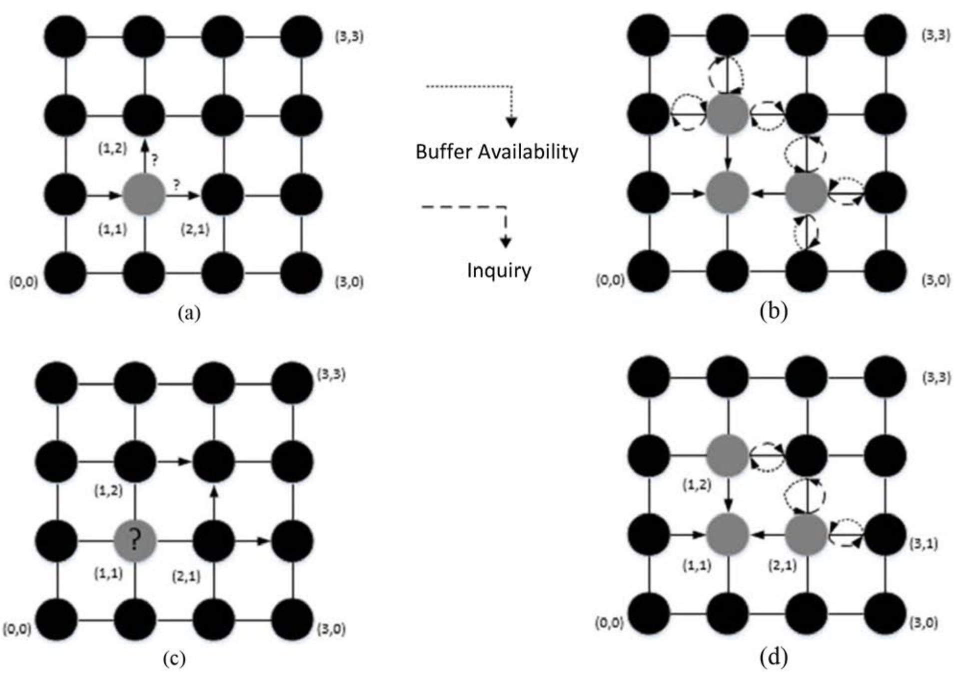


Fig .3. MNoP selection roadmap. a. Condition of hesitation. b. Input buffer availability and number of inquiries of non-adjacent nodes. c. Computation of the adjacent nodes. d. Using of the input buffer and inquiry status of MNoP to make the selection.

algorithm with selecting nodes (2, 1) and (1, 2) as first nodes, to decide appropriate channels that they could lead the flit to arrive at node (3, 3) as a destination node. In fact, node (1, 1) finds out that node (2, 2) and node (3, 1) are on the path guiding to node (3, 3) (Fig. 3c), therefore it will establish its choice of the next appropriate node on the buffer availability and number of inquiries in these nodes (Fig. 3d).

C. MNoP Algorithm

Now, the MNoP selection algorithm functionality is evaluated. Fig. 4 shows the pseudocode of the MNoP selection algorithm executed at node n_c. The input is the number of permissible output channels, AOC, returned by the routing function R, AOC = R(n_c, n_d), where n_d is the destination node of the header flit which should be routed. The output is the selected output channel $sc \in AOC$ (line 1).

```
1.  MNoP_Select(in : AOC, out : sc ) {
2.  scores[] ← 0
3.  counter[][] ← 0
4.  for ch1 ∈ AOC {
5.  node1 ← dest(ch1)
6.  AOC neighbor ← R(node1,nd)
7.  for ch2 ∈ AOC neighbor {
8.  if MNoPData in[ch1].available[ch2]
9.  score[ch1]+ =
10. ( MNoPData in[ch1]. free slots[ch2] * ( 2 ) ) -
    counter[node1][ch2]
11. if(counter[node1][ch2] <=2)
12. counter[node1][ch2]++
13. else
14. counter[node1][ch2] = 0
15. }
16. }
17. sc ← ch st score[ch] = max(scores[])
18. }
```

Fig. 4. MNoP algorithm pseudocode

First, we define scores array for storing the score of each channel and a counter 2D array with two indexes for candidate nodes and ports number (lines 2, 3). For each candidate output port (line 4), the current node, nc, computes the number of MNoP nodes (lines 5-6) to inspect the availability of their input buffers and number of inquiries (we have shown with dest(ch) the destination node linked to channel (ch)). Using a score technique, the score of a candidate destination is increased for each MNoP with available capacity in a input buffer that is not reserved and the number of inquiries is decreased from free slots to affect this parameter in final choice and the number of free slots is multiplied in 2 for preventing negative value (lines 8-9-10). For counting number of inquiries in each node we use a counter and the maximum value is 2 because in two hop counts or two clock cycles it can count to this amount and if it increases from 2 the counter will be reset (lines 11,12,13,14). Finally, the channel with the highest score is selected (line 17). If more than one port has the same maximum score, a random function choose an optional channel.

IV. EXPERIMENTAL RESULTS

A. Simulation Environment

In this section, Noxim [20] as a flit-level simulator in on-chip networks is exploited. We have selected "Hotspot", "Butterfly" and "Shuffle" traffic patterns in simulations in order to evaluate the intended selection strategy. It is necessary pointing out that the proposed selection function (MNoP) is compared to similar selection function, "Neighbor-on-Path" in factors such as latency and energy consumption. In order to provide an extensive size of packet injection rates in simulating the scenarios, the packet injection rates is assumed from 0.05 to 0.4 in one scenario and from 0.08 to 0.09 and 0.1 to 0.11 in the two other scenarios. For evaluating simulation results, the size of input buffer in every node is set to 4 flits. Every scenario has been simulated 10 times in each packet injection rate for achieving more exact outcomes and the average amount is chosen through the computed results. All the communication links have been adjusted to unicast. At the present time we explain each scenario precisely, with its particular features and next, the simulation outcomes and their analysis will be explained for each traffic pattern.

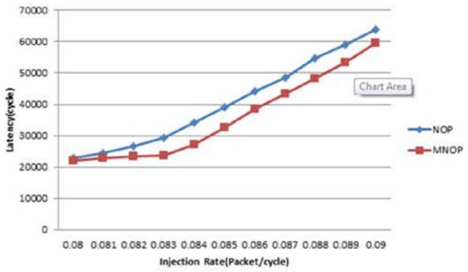


Fig. 5a. Latency result, when using North-last routing under Butterfly traffic pattern

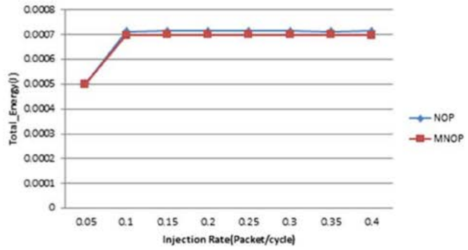


Fig. 5b. Energy result, when using North-last routing under Butterfly traffic pattern

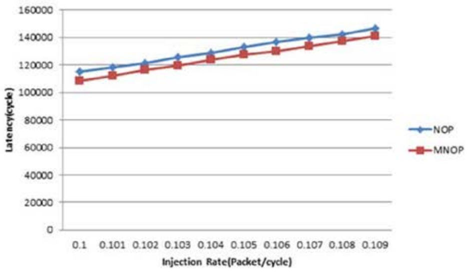


Fig. 5c. Latency result, when using West-first routing under Butterfly traffic pattern

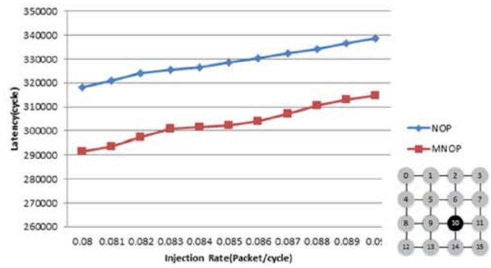


Fig. 5d. Latency result, when using West-first routing under Hotspot traffic pattern, node 10

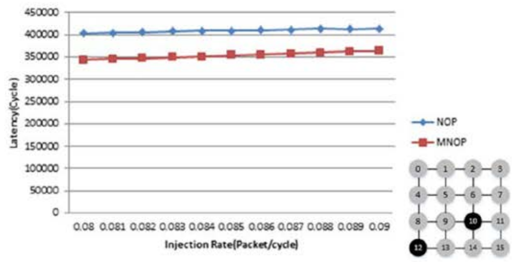


Fig. 5e. Latency result, when using West-first routing under Hotspot traffic pattern , nodes (10,12)

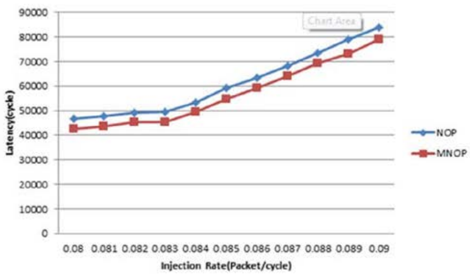


Fig. 5f. Latency result, when using West-first routing under Shuffle traffic pattern

Fig. 5. Latency and Energy Results

B. Simulation Results for Butterfly Traffic Pattern

For the initial scenario, a 4×4 regular 2D mesh topology is supposed. Also, two selection functions, including NoP, and MNoP are assessed for North-last routing algorithm under Butterfly traffic pattern.

Each node transmits flits to adjacent nodes with equivalent probability. Results referred to packet latency are demonstrated in Fig. 5.a. As simulation results show, the intended selection strategy has achieved better latency results in comparison to NoP selection function, after the saturation point when the packet injection rate goes over 0.082. The reason for this improvement can be because of having a better load balancing in network. Regarding energy consumption, Fig. 5.b demonstrates that there is an acceptable improvement in energy consumption between two selection functions and so, the energy consumption of the proposed selection function is satisfactory and improved. In Fig.5.c the West-first routing algorithm with Butterfly traffic pattern is analyzed. Also, the packets injection rate is changed (0.1, 0.11). As it can be seen, the proposed approach outperforms in latency outcomes in comparison to NoP selection function.

C. Simulation Results for hotspot traffic patterns

For the next scenario, a 4×4 regular 2D Mesh topology has been assumed, and the routing algorithm is West-First, but the traffic pattern has been adjusted to Hotspot. Fig. 5.d demonstrates latency results with just one hotspot node (10) and possibility 0.5, comparing NoP selection function, and as obtained results show that in another time the proposed approach outperforms the other selection function in factors such as latency. The energy result is also satisfactory and it even outperforms in comparison to NoP selection function similar to previous scenario results. The reason might be that we are evaluating with a congestion-aware traffic scenario. Fig 5.e demonstrates the latency results with two hotspot nodes (10,12) and possibility 0.5, comparing NoP selection functions , and also it can be seen that once again the proposed selection function performs better than the other function in terms of latency. This proves that if the number of interior hotspot nodes increases the network with the proposed selection function will show a better improvement in results. Ultimately, it also causes more traffic distribution in network load.

D. Simulation Results for Shuffle Traffic Patterns

For the final scenario, a 4×4 regular 2D mesh as the topology and two selection functions, including NoP, and MNop are analyzed for West-first routing algorithm under Shuffle traffic pattern. Every node sends data to all other nodes with equal probability. The results related to packet latency are demonstrated in Fig. 5.f. As it can be seen, the proposed selection function has obtained better latency results in comparison to NoP selection functions, which the reason could be because of having a better traffic distribution in network.

V. CONCLUSION AND FUTURE WORKS

In this paper, a contemporary output selection strategy is proposed, named "MNop" that could be used with adaptive routing algorithms. The general aim is handling the condition of hesitation when the routing function returned more than single output port as nominees in addition to improve the latency overhead generated by the flits remaining to attain a occupied output port. One of the imperative anxieties when upgrading routing in NoC, is refraining the scenarios from congestion. This can be happen by creating a congestion-aware output selection function in order to balance traffic load and minimize the contention as much as possible, in order to obtain better latency results. Simulation results have indicated that when applying the proposed selection function to West-First and North-last routing algorithms under Butterfly , Hotspot and Shuffle traffic patterns, it outperforms in comparison to the NoP selection function in latency results, and an improvement of 20% can be obtained in the best situation, while achieving an improvement in energy results. For the future work, a reset signal will be define that each two clock cycles will reset the counters in order to be more precise in counting number of inquiries. Additional developments such as using different traffic patterns and topologies in order to analyze our proposed selection function under various conditions will be one of the most essential parts of the future works.

REFERENCES

1. W.J. Dally and B. Towles, "Route Packets, Not Wires: On-Chip Interconnection Networks," Proc. ACM/IEEE Design Automation Conf., pp. 684-689, 2001.

2. J.C. Martı´nez, F. Silla, P. Lo´pez, and J. Duato, "On the Influence of the Selection Function on the Performance of Networks of Workstations," Proc. Int'l Symp. High Performance Computing, pp. 292-299, 2000.

3. L. M. Ni and P. K. McKinley. A survey of wormhole routing techniques in direct networks. Computer, 26(2):62–76, 1993.

4. J.Duato,S.Yalamanchili,andL.Ni,Interconnection Networks: An Engineering Approach. Morgan Kaufmann, 2002.

5. G.-M. Chiu, "The Odd-Even Turn Model for Adaptive Routing," IEEE Trans. Parallel and Distributed Systems, vol. 11, no. 7, pp. 729- 738, July 2000.

6. J. Duato, et al., Interconnection networks : an engineering approach. San Francisco, CA: Morgan Kaufmann, 2003, pp. 87.

7. C. Chen-Ling and R. Marculescu, "Contention-aware application mapping for Network-on-Chip communication architectures," in Computer Design, 2008. ICCD 2008. IEEE International Conference on, 2008, pp. 164-169.

8. G. Ascia, et al., "Implementation and Analysis of a New Selection Strategy for Adaptive Routing in Networks-on-Chip," Computers, IEEE Transactions on, vol. 57, pp. 809-820, 2008.

9. B. Niazmand, M. Reshadi, and A. Reza," PathAware: A Contention-aware Selection Function for Application-specific Network-On-Chips," in NORCHIP, 2012, pp. 1–6.

10. D. Salemi, et al., "Power-aware selection policy for networks on chip," in Design & Technology of Integrated Systems in Nanoscale Era (DTIS), 2011 6th International Conference on, 2011, pp. 1-4.

11. M. Palesi, et al., "Bandwidth-aware routing algorithms for networks-on-chip platforms," Computers & Digital Techniques, IET, vol. 3, pp. 413-429, 2009.

12. A. Dana, N. Salehi, "Congestion aware routing algorithm for mesh network-on-chip platform," in Indian Journal of Science and Technology, Vol. 5, No. 6, pp. 2822-2830, June 2012.

13. Wu, Dong, Al-Hashimi, Bashir M. and Schmitz, Marcus T. (2006) Improving Routing Efficiency for Network-on-Chip through Contention-Aware Input Selection. In, 11th Asia and South Pacific Design Automation Conference (ASP-DAC 2006).

14. W. Trumler, S. Schlingmann, T. Ungerer, J. Bahn, and N. Bagherzadeh. Self-optimized Routing in a Network on-a-Chip. In M. Hinchey, A. Pagnoni, F. Ram- mig, and H. Schmeck, editors, Biologically-Inspired Collaborative Computing, volume 268 of IFIP International Federation for Information Processing, pages 199–212. Springer Boston, 2008.

15. P. Gratz, et al., "Regional congestion awareness for load balance in networks-on-chip," in High Performance Computer Architecture, 2008. HPCA 2008. IEEE 14th International Symposium on, 2008, pp. 203-214.

16. S. Azampanah, et al., "LATEX: New Selection Policy for Adaptive Routing in Application-Specific NoC," in Parallel, Distributed and Network-Based Processing (PDP), 2012 20th Euromicro International Conference on, 2012, pp. 515-519.

17. S. Ma, et al., "DBAR: an efficient routing algorithm to support multiple concurrent applications in networks-on-chip," SIGARCH Comput. Archit. News, vol. 39, pp. 413-424, 2011.

18. L. Shu-Yen, et al., "Traffic-and thermal-aware routing for throttled three-dimensional Network-on-Chip systems," in VLSI Design, Automation and Test (VLSI-DAT), 2011 International Symposium on, 2011, pp. 1-4.

19. H. Hsien-Kai, et al., "Regional ACO-based routing for load-balancing in NoC systems," in Nature and Biologically Inspired Computing (NaBIC), 2010 Second World Congress on, 2010, pp. 370-376.

20. F. Fazzino, M. Palesi, and D. Patti, "Noxim: Network-on-Chip simulator," http://noxim.sourceforge.net.