Buffer Depth and Traffic Influence on 3D NoCs Performance

Yan Ghidini[1], Thais Webber[2], Edson Moreno[1], Fernando Grando[1], Rubem Fagundes[2], César Marcon[1]

[1]Faculty of Informatics / [2]Faculty of Electrical Engineering
Pontifical Catholic University of Rio Grande do Sul (PUCRS)
Av. Ipiranga 6681, Porto Alegre, Brazil
yan.souza@acad.pucrs.br

Abstract—**3D NoC-based architectures have emerged to reduce the network latency, the energy consumption and total area in comparison to 2D NoC topologies. However, they are characterized by various trade-offs with regard to the three dimensional structure and its performance specifications. In this paper, we present a 3D NoC mesh architecture called Lasio, whose latency and the throughput achieved, for both network and application, are evaluated considering two types of traffic patterns, varied buffer depth and a range of packet sizes. Cycle-accurate simulations demonstrated that there is a high impact of buffer depth and packet size on the NoC latency and on the application latency. Applying an appropriate buffer depth, for several sizes of packets, the application latency is reduced and throughput is increased.**

Keywords - 3D NoC, buffer depth, latency, throughput.

I. Introduction

Networks-on-chip (NoC) has emerged as a promising interconnection architecture for multiprocessor system-on-chip (MPSoC) platforms. In this new packet-based communication networks, dozens of Processing Elements (PEs) are integrated together with large amounts of embedded memory on a single chip. The NoCs paradigm is proposed as a promising communication platform because of scalability, better throughput and reduced power consumption [1]. However, NoC-based architectures are characterized by various trade-offs with regard to structural characteristics, performance specifications, and application demands.

Two-dimensional NoCs are natural choices of communication infrastructure for the majority of actual chip fabrication technologies. However, wire delay and power consumption are dramatically increasing even when using this kind of topology. In this sense, three-dimensional (3D) NoCs have emerged to reduce the length and number of global interconnections and the number of hops that packets must pass through, and consequently, decreasing the network latency [2]. 3D NoCs offer an opportunity to continue the performance improvements using CMOS technology, with smaller form factor, higher integration density, and support for the realization of mixed-technology chips [3]. In 3D integration technology, multiple layers of active devices are stacked above each other and vertically interconnected using *through-silicon-vias* (TSVs). The major advantage of this emerging technology compared to 2D designs is the inherent reduction in wire length. 3D integration enables NoCs to considerably reduce the network diameter and overall communication distance, enhance communication performance and reduce power consumption [4]. 3D NoC design takes into account architectural issues such as topology, type of interconnection between layers, routing and switching algorithms, buffers depth specification, often according to the application demands or specific design costs. In this work, we prototyped Lasio for latency and throughput evaluations considering different traffic scenarios. Lasio, which is a 3D NoC mesh architecture, is an extension of Hermes 2D NoC [5]. Our aim is to evaluate the significant impact of buffer depth and packets size, on application and network latencies and throughput. In this paper, we characterize the performance of the Lasio architecture in the presence of two scenarios of uniform traffic.

The paper is organized as follows. Section II discusses related works on 2D versus 3D NoC design concerning application requirements, input traffic and the set of experiments used to evaluate these communication architectures. Section III describes the architecture of Lasio, which is the 3D NoC mesh used for our experimental purposes. Section IV presents the experimental setup containing the input traffic scenario, as well as the latency and throughput evaluations. Section V contains the experimental results related to the Lasio implementation under two different traffic scenarios, varying packets size and router input buffer depth. In our analysis we consider not only the NoC latency and the NoC throughput, but also the latency and throughput of the application. Finally, in Section VI we conclude our contribution.

II. Related Works on 3D NoC Evaluation

Current MPSoCs are normally implemented targeting 2D communication architectures. However, the 2D technology has a number of limitations that affect various design requirements. The 3D NoC technology, where multiple communication layers are stacked and connected by a vertical interconnecting schema (e.g. TSV), emerged as an improvement of 2D NoC with respect to power dissipation, energy consumption, transistors density and latency. However, 3D technology brings new design challenges, and many benefits and drawbacks that are not yet entirely known. In this sense, the following related works compare 3D NoC topologies and 3D with its 2D counterpart according to design requirements.

Pavlidis and Friedman [2] proposed analytical models for the zero-load latency and power dissipation estimation, aiming to compare 2D mesh NoC with 3D counterparts. Their experiments show an improvement up to 40% in latency and up to 62% on energy consumption when using 3D NoCs.

978-1-4673-2789-3/12/$31.00 ©2012 IEEE

Park et al. [6] have developed MIRA, which is a 3D mesh NoC topology with additional express channels used to improve the communication. Experiments with random uniform traffic and seven real workloads proved the efficiency of MIRA to minimize the NoC temperature, and reduce the energy consumption and latency when compared to traditional 2D and 3D mesh NoCs.

Feero and Pande [7] have used synthetic localized and uniform traffic to evaluate the performance of 3D NoC architectures. Their study demonstrates the 3D NoC superior functionality in terms of throughput, latency, energy consumption, and wiring area overhead when compared to 2D counterpart implementation.

Xue et al. [8] have developed THIN (a new 3D NoC topology) emphasizing its performance in comparison to other 3D NoCs. They also compared 3D NoCs on the number of links and diameter, as well as against 2D counterparts. Experiments with some synthetic traffic patterns (random, bit reverse and transpose) and several data injection rates show the improvement of THIN in latency and energy consumption.

Agyeman et al. [9] proposed a heterogeneous NoC architecture, which combines 2D routers and 3D NoC-bus hybrid router architectures in 3D mesh topology. Experimental results show that with low latency penalty their approach may save significant area with efficient energy consumption.

Zia et al. [10] evaluated the energy consumption on large MPSoCs for several NoC topologies, concerning 2D NoCs and their 3D counterparts. They also implemented a 3D clos NoC (CNoC) trying to evaluate its scalability and energy efficiency. To evaluate CNoC and others topologies, the experiments take into account the influence of the number of nodes and some router types on the minimization of energy consumption and latency. In addition, the NoC energy consumption has been also evaluated according to flit (minimal logic unit used for flow control) size and the injection rate. They conclude with the efficiency of the proposed CNoC showing that, for all experiments, the 3D NoCs are more energy and latency efficient than 2D counterparts.

TABLE I - RELATED WORK SUMMARY.

Work	NoC	Requirements	Traffic	Experiments
[2]	3D and 2D mesh	latency, energy consumption	analytic models	number of nodes
[6]	3D and 2D mesh	latency, temperature energy consumption	random and seven real applications	injection rate
[7]	3D and 2D mesh	throughput, latency, wiring area, energy consumption	several uniform and localized traffic	injection rate, number of 3D layers
[8]	3D and 2D mesh, torus	latency, energy consumption	transpose, random and bit reverse	number of nodes, injection rate
[9]	2D-3D hybrid mesh	latency, energy consumption, area	random	injection rate
[10]	3D and 2D clos, mesh, ftree, bfly	latency, energy consumption	uniform using Bernoulli process	number of nodes, injection rate, flit size
[11]	3D bus-NoC hybrid mesh	power dissipation, area, latency	random, hotspot NED, video	injection rate
This	3D and 2D mesh	latency, throughput, NoC occupancy	all-to-all and complement	injection rate, flit size, buffer depth

Rahmani et al. [11] have experimented some synthetic (uniform, hotspot, and negative exponential distribution - NED) and real (video conference) benchmarks to demonstrate the power dissipation, packet latency and area consumption efficiency of a proposed bus-NoC hybrid approach for 3D mesh topology (with others 3D hybrid approaches).

Our study extends these previous works (summarized on Table I), focusing on buffers exploration according to traffic characteristics. We show that 3D topologies are able to achieve better performance than their 2D counterpart, but most importantly we show how a 3D architecture could be improved according to the packets size and traffic behavior.

III. Lasio 3D NoC Architecture

Lasio is a 3D mesh NoC that was developed having Hermes 2D NoC [5] as base architecture, i.e., Lasio has the same types of mechanisms and resources of Hermes, supporting more ports to enable 3D communication. The Lasio characteristics are described in the subsequent sections.

A. Lasio Topology

NoC topology is basically defined by the connection structure of the routers. A direct topology is the one where each router has a set of bidirectional ports linked to other routers, and one port linked to a local PE. One of the well-known direct 2D NoC architectures is the 2D Mesh. This architecture consists of $m \times n$ mesh of routers interconnecting the PE placed along with them. The straightforward extension of this popular planar structure is a 3D symmetric NoC, which adds two additional physical ports to each router; one for *top* and other one for *bottom*, where all links have the same length. In the Lasio NoC design it was used direct 3D mesh topology to facilitate the placement of routers and PEs, as well as the routing channels between routers, simplifying the routing algorithm implemented in the control logic. Besides, in the mesh topology used in this work, each router has a unique address in the network expressed in XYZ coordinates, and a different number of ports, depending on its position with regard to the limits of the network, as shown in Figure **1**.

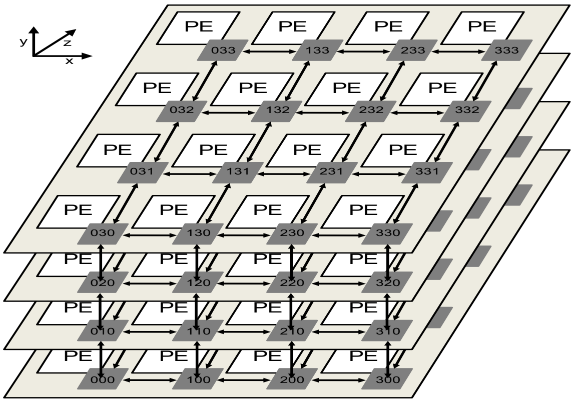

Figure 1 - The mesh-based 3D NoC architecture.

B. Router Interface

Figure **2** illustrates a bidirectional link between two routers in Lasio. The output port is composed by the following signals: (i) *clock_tx* that synchronizes data transmission; (ii) *tx* that controls the data availability; (iii) *data_out*, which is a bus containing data to be sent; and (iv) *credit_i*, which is a control signal that indicates the buffer availability. In addition, the input port is composed by the following signals: (i) *clock_rx*; (ii) *rx*; (iii) *data_in*; and (iv) *credit_o*, which are the counterpart of the output port signals, respectively. As a consequence, each bidirectional link has 6 control signals and 2×flit data signals[1]. Moreover, Lasio implements vertical links applying TSV technology, for instance, in Figure **2** Router 121 is connected with the Router 131 through a TSV bidirectional link.

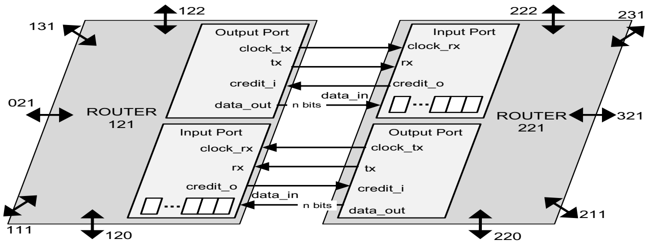

Figure 2 - Lasio bidirectional link (e.g. of routers 121 and 221 - Figure 1).

C. *Router Architecture – Arbitration and Switching*

The main objective of an on-chip router is to provide end-to-end communication between PEs. The Lasio router contains control logic responsible for routing and arbitration, a crossbar structure, and seven bidirectional communication ports connected to other routers and PEs, as illustrated in Figure **3**.

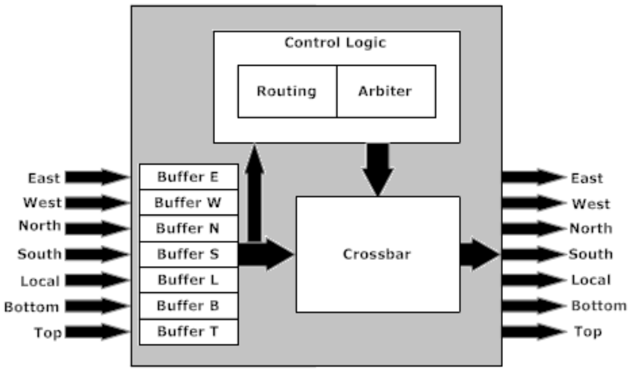

Figure 3 - The Lasio router architecture.

Five ports are dedicated to connections made within each layer (Local, North, South, East and West). Two other ports (Top and Bottom) ensure the communication between adjacent layers. The Local port establishes a communication between the router and its corresponding PE, while the remaining ports are connected to neighboring routers. Each communication port includes input and output channels which has a buffer working as circular FIFO with configurable size for temporary data storage, which is used when a the routing path is congested by other packets. Also, the crossbar module is responsible for ports switching. It indicates which ports are connected then verifying data to be transmitted and the ports availability.

The packets employed in the transmission are composed by a target address field, a size field, and a payload. Size field is the number of flits contained in the payload. Figure **4** shows the basic composition of a Lasio NoC packet.

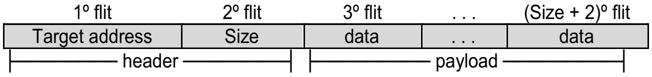

Figure 4 - Lasio NoC packet.

Lasio implements XYZ routing algorithm, which is an extension of the XY routing algorithm exploited in 2D NoCs. This routing algorithm is deadlock free and enables small area of implementation. When a router receives a header flit, the arbitration is executed, and if the incoming packet request is granted, the XYZ routing algorithm is performed to connect the input port data to the correct output port. From the source router to the target router, packets are routed firstly in X, after in Y and then in Z coordinates, respectively, passing through several buffers and ports. If the chosen output port is busy, all subsequent flits are blocked in the input buffer, and the request remains active until the connection with the port is established. When the target port is free, the arbitration algorithm takes place to decide which request will be served (in case of concurrent requests), establishing a connection between an input port and an output port.

The arbiter uses a dynamic rotating policy that prioritizes the packet routing on the input port. In other words, the arbitration is implemented using Round Robin algorithm. This method ensures that all incoming requests are processed, preventing starvation phenomenon. The arbitration logic takes four clock cycles to address a routing request. This additional time is necessary for the execution of the routing algorithm. If the routing algorithm is able to establish a connection to the desired output port, the input port requires to the arbiter a new routing request.

There are three vectors called *in*, *out* and *available* located in a switching table, which are explored during the execution of the routing algorithm. The *available* vector is used to indicate the availability of a given output port, meaning if the port is transmitting a packet (i.e. busy) or if not (i.e. free). When there is a transmission request from an input port, the routing policy tries to find an available output port. In this case, all three vectors are updated. Otherwise, the input packet remains contained in the input buffer. The *available* vector position related to the output port is set as busy, while the *in* and *out* vectors are interlaced. The *in* vector indicates to which output port the packet is being routed, while *out* indicates to which input port the packet is coming from.

Figure **5** exemplifies the router switching process, e.g. the North port has its output port set as busy because it is transmitting a packet from West input port. At the same time, the North input port is assigned to the Top port. When the packet transmission is finished, the available vector is updated, i.e. set to free.

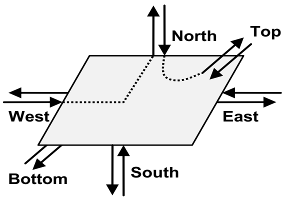

Figure 5 - Example of two simultaneous connections in the router.

TABLE exemplifies the switching illustrated in Figure **5**.

TABLE II - EXAMPLE OF LASIO 3D NOC SWITCHING TABLE.

	Port name						
	East	West	North	South	Local	Bottom	Top
available	free	free	busy	free	free	free	busy
in	-	North	Top	-	-	-	-
out	-	-	West	-	-	-	North

D. *Lasio Switching and Flow Control*

Lasio NoC implements wormhole switching method be-

[1] *The flit size of Lasio is equal to the phit size.*

cause of some advantages as: (i) the need for smaller buffers for storing data, and (ii) low-latency communication. The wormhole mode implies dividing packets into flits (please refer to Section C). The flit size for the Lasio infrastructure is customizable, and the number of flits in a packet is limited to $2^{(\text{flit size in bits})}$.

Furthermore, Lasio NoC utilizes credit-based flow control, which is an optimized communication mechanism, since it may consume few clock cycles to perform a flit transmission. This method utilizes FIFO buffers with customizable size at the receiver input, and a return line to the transmitter informing if there is available space in the buffer. This information can be interpreted by the transmitter as a credit, thus it just sends data if a credit is available. This approach does not allow packets to be discarded because one transmission between routers starts only after verifying if data will be received. This verification is performed in two steps: the receiver sends to the transmitter via *credit_o*/*credit_i* (Figure **2**) signal information indicating credits availability and the transmitter sends data only when there is credit available.

IV. Experimental Setup

A. Traffic Scenarios

Latency and throughput, which are metrics evaluated in this work, are strongly dependent of communication pattern. However, when choosing two synthetic traffic scenarios (*all-to-all* and *complement*) whose characteristics are determinism and uniformity of packet load, evaluations of latency and throughput can be made independent of the communication pattern.

- Complement

In the *complement* traffic scenario packets are generated and injected into the network simultaneously, then the router located in the first position inside the NoC sends packets to the router located in the last position inside the NoC; the router located in the second position inside the NoC sends packets to the router located in the next-to-last position, and so on. Basically each network router sends packets to the router located at its complementary position as Figure **6** illustrates.

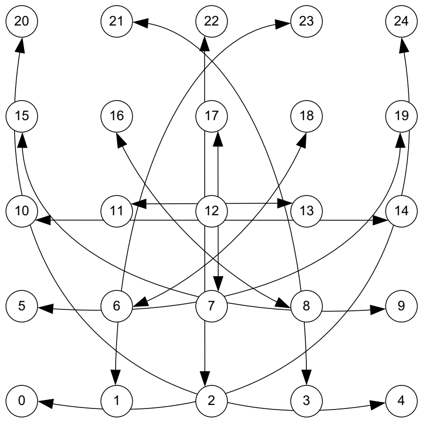

Figure 6 - Example of complement traffic scenario for 5x5 2D mesh NoC.

- All-to-all

In this traffic scenario all routers send the same quantity of data (uniform packet load) in a deterministic way to all others routers, except to itself. Firstly, each one of the routers sends simultaneously a packet to router 0. Then, in the same way, each one of the routers sends another packet to router 1, and so on. This model of traffic is used to cover as many traffic and blocking situations as possible, since a large number of packets are traveling on the network simultaneously. Despite the *all-to-all* traffic scenario not be the best approach when dealing with real applications, since the communication destinations change frequently, such traffic scenario allows encounter any shortcomings in the communications infrastructure more easily [12].

B. Lasio Architecture Assumption

Although vertical links are normally considered a bottleneck in 3D NoC design, mainly for implying links with more area, and sometimes mechanisms to serialize and deserialize the communication. This work assumes that inter-layer (horizontal links) and intra-layer (vertical links) hops are indistinguishable, which means the hops between layers (Z coordinate) and the hops within each layer (coordinates X and Y) have the same cost in terms of latency and throughput.

C. Performance Evaluation

In this work we analyze latency and throughput, for both network and application, as performance evaluation metrics. Results were achieved by cycle-accurate simulations of the VHDL RTL description of Lasio 3D NoC, performing several simulations with varied packet sizes and buffer depth.

- Latency analysis

Packets latency metric can be observed in different ways, as shown in Figure **7**. The communication latencies presented here are not limited to the packets transmission delay into the NoC, but they also consider the packets that are delayed to enter into the NoC. Following we present some explanations to facilitate the differentiation of the transmission latencies according to the distribution of packets injection and packets reception.

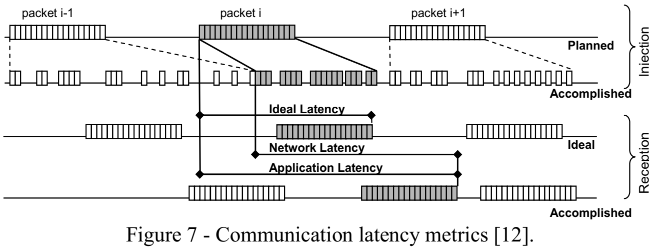

Figure 7 - Communication latency metrics [12].

The *planned injection* is the moment that a packet is able to be injected into the NoC. In our simulation scenarios all packets are specified in an input text file with its planned injection time. *Accomplished injection* considers the exact insertion timing of a packet into the NoC, which may be different from that defined in *planned injection*, due to the occurrence of contentions. The *ideal reception* represents the estimated time of packets delivery. The *accomplished reception* shows the real delivery time of packets at their destinations. Figure **7** shows distributions of such injection and reception scenarios. The *ideal latency* is the minimum number of clock cycles that a packet needs to reach its destination. This value is obtained from the difference between the *planned injection* time of the packet and the expected delivery time of the same packet.

In this paper, the concept of **network (NoC) latency** is related to the transmitting delay of a packet from source to destination, which can be influenced by other packets competition for NoC resources (e.g. channels and storage queues). On the other hand, **application (App) latency** expresses the time spent between the moment a packet is created by the application and the moment the packet is consumed by the target node. Application latency illustrates the most important impact on the ideal performance of a communication, since it is computed as the difference between the *planned injection* time of packets and their exact delivery moment at destination [12]. Both **NoC latency** and **App latency** are assumed as performance metrics for comparison in this work.

- Throughput analysis

Another parameter considered to evaluate NoCs performance was the packet throughput. Here, similarly to latency, we consider both **network (NoC) throughput** and **application (App) throughput**. While the **NoC throughput** metric evaluates the NoC capacity on packets transmission, the **App throughput** considers in addition the packets delay before entering on network. Both packet throughput metrics are influenced by the buffer depth, traffic scenario, packet size and other NoC resources competitions.

V. LASIO PERFORMANCE EVALUATION

This section presents an architecture evaluation considering network and application latencies and throughput to a specific 3D NoC called Lasio. In such evaluation, *all-to-all* and *complement* traffic scenarios were used with injection rate of 800 Mbps. The buffer depth is set to 2^n flits, where *n* ranges from 2 to 10. To provide a thorough comparison, we prototype two NoCs, both containing 64 tiles and routers in a square and cubic format: Hermes 2D NoC [5] is 8x8 mesh and Lasio 3D NoC is 4x4x4 mesh. Moreover, both does not contain virtual channel and are credit based control flow. We varied the packet size in some experiments from 5-flit packets to 1024-flit packets (considering each flit is 16 bits width).

A. Network and Application Latencies

The first approach in analyzing the Lasio 3D NoC performance is to compare it to its 2D counterpart. Both NoCs were initially experimented with packets of 5-flit size and *all-to-all* traffic pattern.

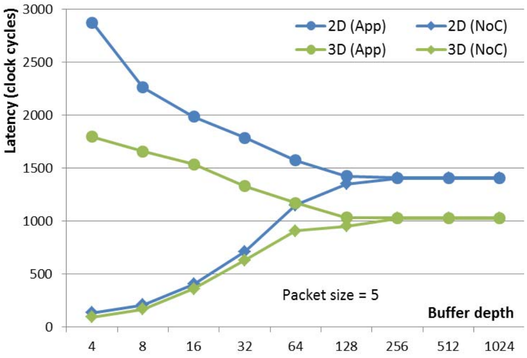

Figure 8 – NoC and App latencies comparison between Lasio 3D NoC (4x4x4 mesh) and Hermes 2D NoC (8x8 mesh).

When comparing 2D and 3D NoCs performance, one should analyze the trade-off existent between buffer depth and topology. Observing Figure **8**, for 128-flit buffer depth or higher, the difference between App latency and NoC latency tends to zero in both 2D and 3D NoCs. This behavior indicates that the latencies are no longer influenced by buffer depth, thus the use of bigger buffers is not necessary for transmitting 5-flit packets. Moreover, these preliminary results highlight that when applying an appropriate buffer depth, the App latency is reduced, e.g. in our experiments up to 3.4 times for 2D topologies and 2.3 times for 3D topologies.

Figure **9** shows the behavior of the NoC latency and the App latency under *all-to-all* traffic pattern, for different buffer depth, and k-flit packet sizes (*k* varies from 5 to 64). It is a fact that the maximum performance is achieved when App latency is equal to NoC latency, since in this case packets that are injected by the application are promptly consumed by the NoC. For instance, 32-flit packet size has as optimum a 512-flit buffer depth.

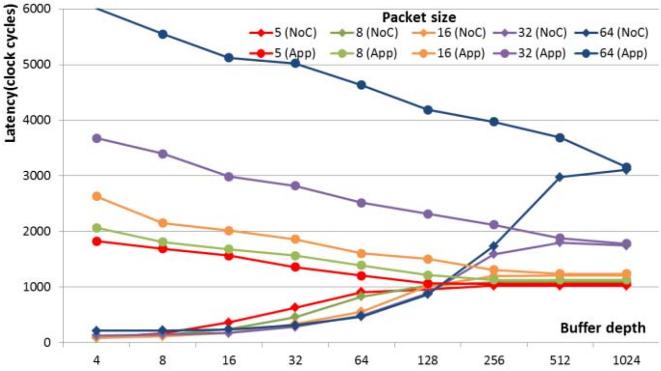

Figure 9 - NoC latency versus App latency for different buffer depth and packet sizes.

Observing Figure **9**, we can notice that greater is the packet size, greater is the observed App latency, the same behavior is not observed for NoC latency, since it is more dependent on buffer depth and packet size relation. This last behavior may be justified by the influence of routing and switching strategies. In both comparisons, Figure **8** and Figure **9**, the increase of buffer depth implies in a gradual slight decrease of App latency; however it compromises NoC latency. This phenomenon can be explained by the fact that the increase of buffers enables a better distribution of packets into the NoC reducing contention and, in average, approximating the payload flits to the target routers.

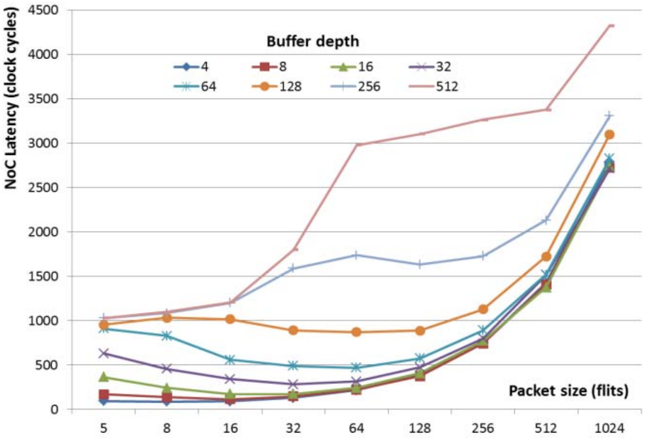

Figure 10 - NoC latency varying packet sizes for different buffer depth.

Figure **10** present the NoC latency obtained from experiments with different packet sizes (5-flit to 1024-flit) simulating varied buffer depth (from 4-flit to 512-flit length). It is observable a low NoC latency with smaller packet sizes such as 5-flit until 16-flit, even for 32-flit packet size, combined with buffer depth of 4-flit up to 32-flit. From 64-flit buffer depth there is a substantial increase and variability on the NoC latency. It happens due to large buffers enabling to insert more packets, which are concurring for the same paths in the NoC. As a consequence, while App latency is reduced, the NoC latency is increased.

Figure **11** shows the results obtained for App latency varying packet sizes and buffer depth. App latency tends to minimize slightly according to the packet size, while, the influence of buffer depth is more significant. The reason of that remains on the fact that both imply on the increase of packets concurring for NoC resources.

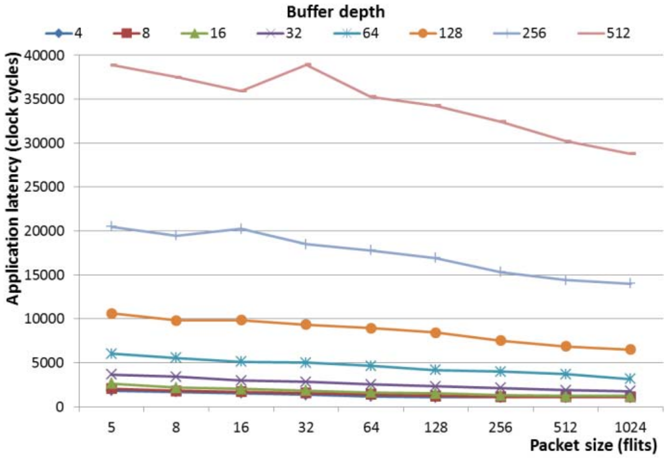

Figure 11 - App latency varying packet sizes for different buffer depth.

Figure **12** shows the traffic influence in NoC latency and in App latency according to different buffer depth. In order to evaluate both latencies, an 8-flit packet size was utilized.

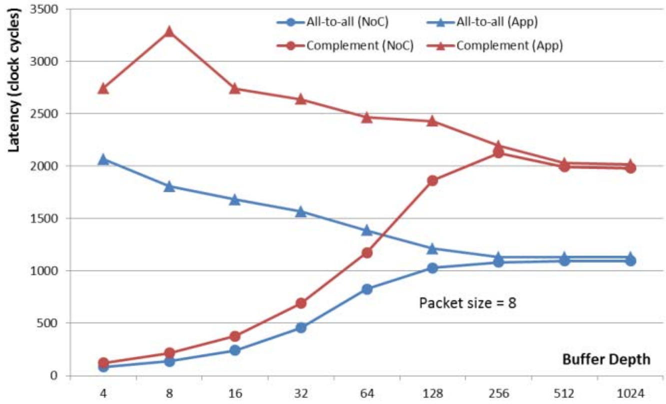

Figure 12 - Traffic influence on NoC latency and on App latency.

Examining the Figure 12, it is possible to observe that, for all buffer depth, both NoC latency and App latency of *complement* traffic are greater than the ones observed in *all-to-all* traffic scenarios. This behavior can be explained by the greater number of hops performed by *complement* traffic pattern. In addition, *complement* traffic implies the same NoC latency and App latency curves behavior pointed in Figure 8 and Figure 9, i.e. the increase of buffer depth decreases App latency and increases NoC latency until a given buffer depth.

B. *Network and Application Throughput*

The throughput of the communication infrastructure generally depends on the traffic pattern, but not only. In these experiments we studied other dimensions such as the relation of buffer depth and packet size under *all-to-all* traffic pattern.

Figure **13** shows that the NoC throughput is directly dependent of the packet size applied, i.e. the NoC throughput is increasing as the packet size augments. This behavior is independent of the buffer depth. However, smaller buffer depth has guaranteed the highest NoC throughput in our experiments. On the other hand, the App throughput remained almost constant for any buffer depth and packet size.

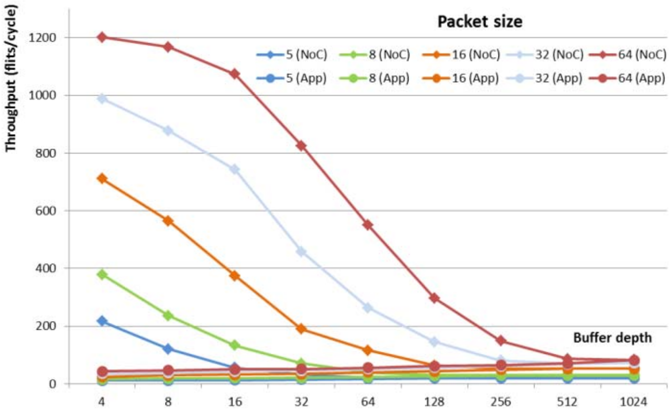

Figure 13 - NoC throughput and App throughput varying packet sizes for different buffer depth.

Looking closely Figure 13, we can observe separately, the evaluations for NoC throughput (Figure 14) and App throughput (Figure 15), for different packet size and buffer depth. Figure **14** shows the results of NoC throughput, having increasing values for throughput augmenting packet size, which is a natural behavior since more flits are being transmitted. Moreover, the highest throughput observed are those related to smaller buffer depth, since fewer packets are actually concurring for resources, then reaching fast their targets.

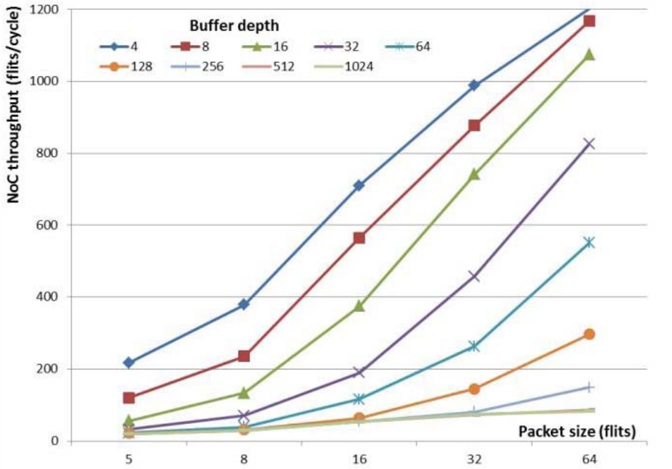

Figure 14 - NoC throughput varying packet size and buffer depth.

Figure 15 shows the App throughput increase according to buffer depth and packet size. Thus, there is a trade-off between the cost to implement greater buffers and the App throughput desired (design requirements).

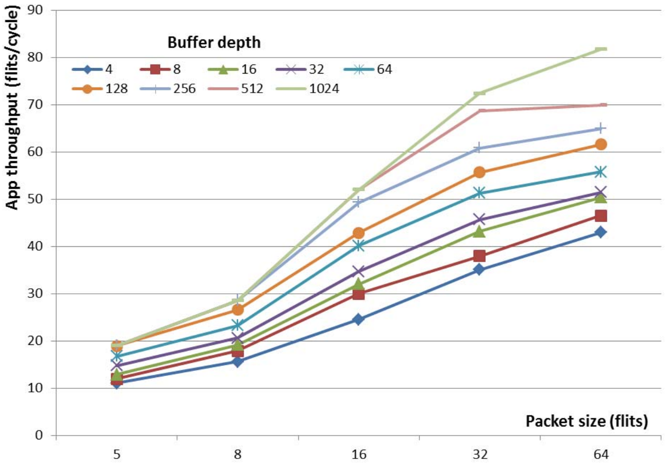

Figure 15 – App throughput varying packet size and buffer depth.

Figure 16 illustrates the traffic influence on both NoC and App throughput according to different buffer depth, considering an 8-flit packet size.

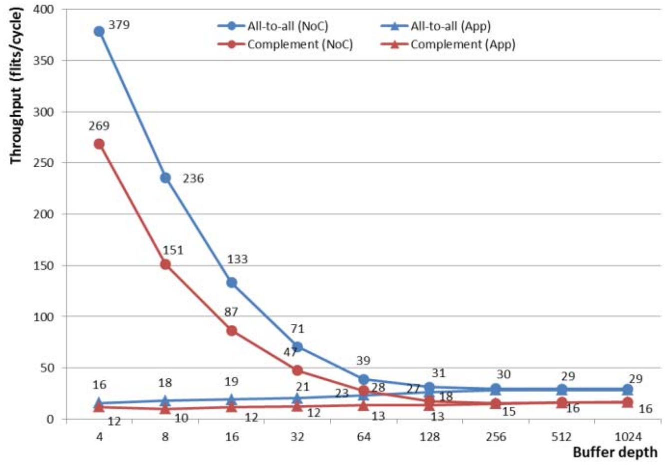

Figure 16 – Traffic influence on NoC and on App throughput.

Observing the Figure 16, NoC throughput is superior to the App throughput in both traffic scenarios regardless buffer depth until 256-flit, when the NoC and App throughput values are approximately the same for each traffic pattern. Moreover, *all-to-all* traffic pattern presents higher throughput than *complement*. Examining the results we have noticed higher NoC throughput for both traffic scenarios when smaller buffers are utilized. On the other hand, the App throughput increases according to the buffer depth, and it stabilizes from 512-flit buffer for both traffic scenarios. This set of experiments enables to detect a more appropriate buffer depth in order to attend some 3D NoC design requirement.

VI. Conclusion

High populated MPSoCs require more efficient communication architectures, to fulfill the PEs traffic demand, which may be provided by a network with 3D technology. This work shows 3D NoC (Lasio), whose implementation is based on 2D NoC (Hermes) [5]. Lasio implements all architectural features of Hermes, except the ones that have to be changed to perform the improvement from 2D to 3D technology, enabling a fair comparison between 2D and 3D NoC topologies.

Our paper explores comparisons of both NoC and application latencies, and both NoC and application throughput according to several buffer depth, packet sizes and two traffic scenarios (*all-to-all* and *complement*). For the selected set of experiments using *all-to-all* traffic scenario, Lasio architecture implementation always minimized the latencies of the packets when compared to Hermes architecture implementation. For instance, the average NoC latency minimization is 25% and the average application latency minimization is 30%.

The paper also explores the influence of several buffer depths and packet sizes on network and application latencies on the Lasio 3D NoC, showing that when applying an appropriate buffer depth both latencies are reduced as well as the throughput is increased.

Acknowledgment

The Authors acknowledge the support of CAPES through research grant (PNPD 058792/2010 and AEX 11392-12-6).

References

[1] A. Jantsch and H. Tenhunen. Network on Chip. Kluwer Academic Publishers, 312p., Jan. 2003.

[2] V. Pavlidis and E. Friedman. 3-D Topologies for Networkson-Chip. IEEE Transaction on Very Large Scale Integration Systems, v.15, n. 10, pp. 1081-1090, Oct. 2007.

[3] L. Carloni, P. Pande and X. Yuan. Networks-on-Chip in Emerging Interconnect Paradigms: Advantages and Challenges. International Symposium on Networks-on-Chip, pp. 93-102, 2009.

[4] A. Rahmani, K. Latif, P. Liljeberg, J. Plosila and H. Tenhunen. Research and practices on 3D networks-on-chip architectures. Proceedings of the IEEE International NORCHIP Conference, pp. 1-6, 2010.

[5] F. Moraes, N. Calazans, A. Mello, L. Möller and L. Ost. HERMES: an infrastructure for low area overhead packet-switching networks on chip. The VLSI Journal Integration, v.38, n.1, pp. 69-93, Oct. 2004.

[6] D. Park, S. Eachempati, R. Das, A. Mishra, Y. Xie, N. Vijaykrishnan and C. Das. MIRA: A Multi-layered On-Chip Interconnect Router Architecture. International Symposium on Computer Architecture (ISCA), pp. 251-261, 2008.

[7] B. Feero and P. Pande. Networks-on-Chip in a Three-Dimensional Environment: A Performance Evaluation. IEEE Transactions on Computers, v. 58, n. 1, pp. 32-45, Jan. 2009.

[8] L. Xue, F. Shi, W. Ji and H. Khan. 3D floorplanning of low-power and area-efficient Network-on-Chip architecture. Microprocessors and Microsystems, v. 35, n. 5, pp. 484-495, Jul. 2011.

[9] O. Agyeman, A. Ahmadinia and A. Shahrabi. Low power heterogeneous 3D Networks-on-Chip architectures. International Conference on High Performance Computing and Simulation (HPCS), pp. 533-538, 2011.

[10] A. Zia, S. Kannan, H. Chao and G. Rose. 3D NoC for many-core processors. Microelectronics Journal, v. 42, n. 12, pp. 1380-1390, Dec. 2011.

[11] A. Rahmani, P. Liljeberg, J. Plosila and H. Tenhunen. An Efficient Hybridization Scheme for Stacked Mesh 3D NoC Architecture. Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP), pp. 507-514, 2012.

[12] E. Moreno et al. Arbitration and Routing Impact on NoC Design. International Symposium on Rapid System Prototyping, pp.193-198, 2011.