Exploration of Alternative Topologies for Application-Specific 3D Networks-on-Chip*

Alexandros Bartzas, Nikolaos Skalis, Kostas Siozios, Dimitrios Soudris

VLSI Design and Testing Center – Democritus Univ. Thrace, 67100 Xanthi, Greece
Email: (ampartza, ns2842, ksiop, dsoudris)@ee.duth.gr

ABSTRACT

Three dimensional (3D) Network-on-Chip (NoC) architectures combine the benefits of new integration technologies with NoC-style interconnection of large number of IP cores in a single package. In this work, we propose a fully software-supported exploration methodology capable of defining pattern-based, alternative, interconnection topologies for application-specific multi-layered 3D NoC architectures. The focus of our exploration is on the number of vertical interconnects (or through silicon vias) connecting grids of different layers, considering the mesh and torus architectures. Existing 3D NoCs assume that every router of a grid can communicate directly with the neighboring routers of the same grid and with the ones of the adjacent layers. We show that this full vertical connectivity is not needed. The exploration methodology is able to evaluate pattern-based 3D topologies and propose the ones that meet the design constraints best. We evaluate the exploration employing and extending the Worm_Sim NoC simulator and feeding it with various types of traffic. In this way, we achieve a decrease in the number 3D routers and in the number of vertical vias, resulting in a decrease in the area occupied by the switch blocks, reducing energy dissipation and paying a negligible penalty in the latency of the 3D NoC.

1. INTRODUCTION

Future integrated circuits will contain billion of transistors, composing tens to hundreds of intellectual property (IP) cores. These IP cores, integrated to devices, will implement emerging complex and demanding multimedia and networking applications being able to deliver rich multimedia and networking services. An efficient cooperation among the IP cores (e.g., data transfers) is needed in order to maximize the available resource utilization. The design of systems composing of such a high number of cores has many challenges.

One challenge is to find an on-chip interconnection network that will be able to connect efficiently the IP cores. Another challenge is to find such an application mapping that will make efficient usage of the available hardware resources. An architecture that will be able to accommodate all these cores, satisfying the need for communication and data transfers is the Network-on-Chip (NoC) architecture [1, 11]. For these reasons Networks-on-Chip are a popular choice when it comes to design the on-chip interconnect for emerging MPSoCs, and are supported from the industry (an example is the Æthereal NoC [9] from Philips). As it is presented in [18] the key design challenges of emerging NoC design are a) the communication infrastructure, b) the communication paradigm selection and c) the application mapping optimization.

The type of the IP cores (their characteristics, capabilities) as well as the topology and interconnection scheme plays an important role on how efficiently an NoC will perform for an application. Furthermore, the application behavior (e.g., data transfers, communication and computation needs) plays an equally important role in the overall performance of the NoC system. For this reason, in order to take full advantage of the hardware resources the NoC architecture should be able to accommodate efficiently the applications' needs providing an application-specific (an application-domain specific) architecture. An overview on what are the costs considerations on the design of NoCs is given at [4].

Up to now NoC designs were limited to two dimensions. But 3D technology exhibits, among others, two major advantages, higher performance and smaller energy consumption [2]. So, due to process / integration technology advancements it is feasible to create NoCs that will expand to the third dimension (3D NoCs). In order to satisfy the demands of emerging systems for scaling, performance and functionality 3D integration is a way to accommodate these demands. For example, a considerable reduction can be achieved in the number and length of global interconnect using three-dimensional integration [12]. A survey of existing 3D fabrication technologies is presented in [3], showing the available interconnection architectures among the layers of 3D integrated circuits and illustrating the research issues in current and future 3D technologies.

In this work we present an exploration methodology of alternative 3D NoC architectures. These architectures are composed of many layers, where each layer is a two-dimensional NoC grid, where the grids are the same for all the layers (composed of elements of the same types). The main objective of the methodology is to find heterogeneous 3D NoC topologies with a mix of 2D and 3D routers and vertical link interconnection patterns that performs best to the incoming traffic. The cost factors we consider are energy consumption, average packet latency and total switch block area, and we compare against a NoC that all the routers

*This paper is part of the 03ED593 research project, implemented within the framework of the "Reinforcement Programme of Human Research Manpower" (PENED) and co-financed by National and Community Funds (75% from E.U.-European Social Fund and 25% from the Greek Ministry of Development-General Secretariat of Research and Technology).

are 3D ones. We have used and extended the Worm_Sim NoC simulator [16], being able to model these heterogeneous architectures and simulate them, gathering information on how they perform. In order to achieve the heterogeneity we use a mix of two- and three-dimensional routers for each layer of the NoC. This use of different routers leads to a "reduced" presence of vertical interconnection links. The methodology evaluates such heterogeneous topologies, targeting mesh and torus ones, for various inputs and shown which ones can handle best the corresponding types of traffic.

The rest of the paper is organized as follows. In Section 2 the related work is described. In Section 3 we present the 3D NoC topologies under consideration, whereas in Section 4 the proposed methodology is introduced. In Section 5 the simulation process and the achieved results are presented. Finally, in Section 6 the conclusions are drawn and the future work is outlined.

2. RELATED WORK

The on-chip interconnection is a widely studied research field and good overviews are [6, 7], illustrating the various interconnection schemes available for present ICs and emerging Multiprocessor Systems-on-Chip (MPSoC). The use of an NoC-type interconnection is able to provide an efficient and scalable infrastructure that is going to be able to handle the increased communication needs.

In order to support NoC design a number of simulators has been developed, such as the Nostrum [15], Polaris [25], XPipes [5] and Worm_Sim [16] using C++ and/or SystemC. To provide adequate input / stimuli to an NoC design, usually synthetic traffic is used. Several synthetic traffic generators exist [10, 22, 24, 26] that are able to provide adequate inputs to the NoC simulators in order to test the communication infrastructure designs.

In [19] it is proposed a methodology that is able to synthesize NoC architectures where long-range links are inserted on top of a mesh network. In this way the NoC is transformed to an application specific one, but it is limited to two dimensions. Li et al. [14] present a mesh-based 3D network-in-memory architecture. They present a hybrid NoC/bus interconnection fabric to efficiently accommodate processors and L2 cache memories in 3D NoCs.

Pande et al. [20] present an evaluation methodology in order to compare the performance and other metrics of a variety of NoC architectures. But, this comparison is made only among two dimensional NoC architectures. The work of Feero and Pande, presented in [8], illustrates that the 3D NoCs are advantageous when compared to 2D ones (with both having the same number of components in total). It is demonstrated that besides reducing the footprint in a fabricated design, three-dimensional network structures provide a better performance compared to traditional, 2D NoC architectures. This works shows that despite the face of a small area penalty, 3D NoCs achieve significant gains in terms of energy, latency and throughput.

In the work of Pavlidis and Friedman [21] various 3D NoC topologies are presented and evaluated. The authors assume 100% vertical interconnection via and their work is focused at the physical level of these silicon vias. Kim et al. [13] present an exploration of communication architectures on 3D NoCs, while we consider a hop-by-hop router connection. These works, both from the physical level as well as adding more communication architectures, such as full 3D crossbar and bus-based communication, are complementary to the one presented here and can be used in order to further extend our methodology.

The main differentiator with the related work is that we do not assume full vertical interconnection (as it is depicted in Figure1), but a heterogeneous interconnection fabric, composed of a mix of 3D and 2D routers. The motivation for this heterogeneous design is the reduced total interconnection network length and the reduced size of the 2D routers have when compared to the 3D ones [8]. Some of the routers of the system have connections with the neighboring ones of the same grid. Whereas, when we say that a router is a 3D one, it means that it has direct, hop-by-hop, connections not only with the neighboring routers belonging to the same grid but also to the ones belonging to the adjacent layers. This difference between two- and three-dimensional routers for a 3D mesh NoC is illustrated in Figure 1, where it is depicted a grid that belongs to a 3D NoC and in that grid are present 2D and 3D routers.

3. ALTERNATIVE VERTICAL INTERCONNECTION TOPOLOGIES

Assuming that we have a 3D NoC and each grid has dimensions $X \times Y$, and in each grid only $K\%$ of the routers can have connections to the vertical direction as well. The available scenarios of how these 3D routers can be place on the grids are:

1. **Uniform:** distribution of the 3D routers over the different layers. Using this scheme we "spread" the 3D routers along every layer of the 3D NoC. In order to find the place of each router we work like this: a) first place the first 3D router of the (x, y) position of each layer, b) then the four neighboring 2D routers are placed in the positions `(x+r+1, y, z)`, `(x-r-1, y, z)`, `(x, y+r+1, z)` and `(x, y-r-1, z)`. The r parameter is defined as:

$$r = \lfloor \frac{1}{K\% - 1} \rfloor \quad (1)$$

   and it represents the number of 2D routers among consecutive 3D ones. In Figure 1(b) is illustrated this scheme, depicting one layer of a 3D NoC, with $K = 25\%$, meaning that $r = 3$.

2. **Center:** All the 3D routers are "gathered" at the center of each layer, as it is depicted in Figure 1(c). Since vertical interconnection links exist only in the center of the layer, in the outer region of the NoC grid the routers are 2D ones, connecting only to the neighboring routers of the same grid.

3. **Periphery:** The 3D routers are positioned at the periphery of the each layer (as it is shown in Figure 1(d)), in a sense the opposite vertical interconnection link to the scheme presented earlier. In this case, the NoC is focused in serving best the communication needs of the outer cores.

4. **Full Custom:** The position of the 3D routers is fully customized matching perfectly the needs of the application with the NoC architecture. This solution fits best the needs of the application, while it minimizes

the occupied area by the switching blocks, by "reducing" the number of vertical and thus the number of 3D routers. However, this full custom solution leads to high design-time cost, a highly not regular design that will not adjust well in a potential change of the functionality, the number of applications that are going to be executed, etc.

In order to perform exploration towards full custom interconnection schemes real applications and/or application traces are needed. In this paper we have used various types of synthetic traffic, so the exploration for full custom interconnections schemes is out of the scope.

In the context of this work we perform exploration towards pattern based vertical interconnection topologies (categories 1-3). We have considered ten different vertical link interconnection topologies. For each of these topologies the number of 3D routers is given (for a $4 \times 4 \times 4$ NoC architecture).

Full: where all the routers of the NoC are 3D ones (as in Figure 1(a)). Number of 3D routers: 64.

By three: It is a pattern based topology with $r = 3$, like the one depicted in Figure 1(b). Number of 3D routers: 44.

By four: Pattern based topology with $r = 4$. Number of 3D routers: 48.

By five: Pattern based topology with $r = 5$. Number of 3D routers: 52.

Odd: In this pattern all the routers belonging to the same row are of the same type. Two adjacent rows never have the same type of router. Number of 3D routers: 32.

Edges: Where the center (dimensions $x \times x$) of the 3D NoC has only 2D routers. Number of 3D routers: 48.

Center: Where only the center (dimensions $x \times x$) of the 3D NoC has 3D routers. Number of 3D routers: 16.

One_side: Where one side (e.g., outer row) of each layer has 2D routers. Number of 3D routers: 48.

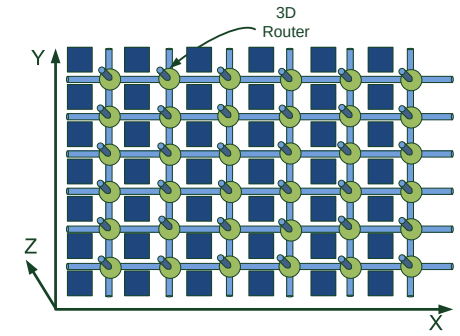
Two_side: Where two sides of all the layers of the NoC have 2D routers Number of 3D routers: 36.

Three_side: Where three of the layer sides have 2D routers. Number of 3D routers: 24.

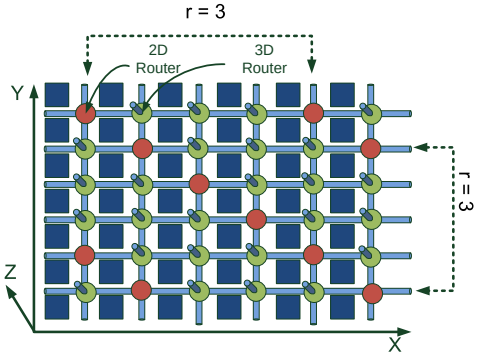
Each of the aforementioned vertical interconnection schemes has advantages and disadvantages and how these schemes perform is based on the behavior of the applications that are implemented on the NoC. As it is explained in the experimental results (Section 5) a wrong choice may diminish the gains of using a 3D architecture.

4. OVERVIEW OF THE EXPLORATION METHODOLOGY

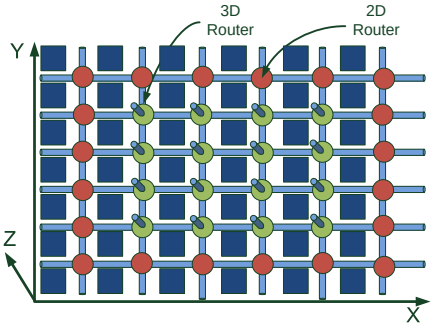
An overview of the proposed methodology is depicted in Figure 2. In order to perform the exploration for alternative topologies of 3D NoC architectures, we used as a basis the Worm_Sim NoC Simulator, utilizing wormhole switching [17].



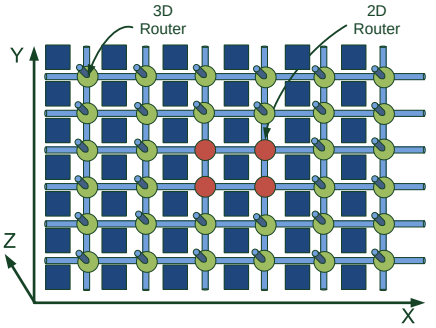
(a) Full vertical interconnection (100%) for a 3D NoC.



(b) Uniform distribution of vertical links.



(c) Positioning of vertical links at the center of the NoC.



(d) Positioning of the vertical links at the periphery of the NoC.

(e) Legend:

Figure 1: Positioning of the vertical interconnection links, for each layer of the 3D NoC (6×6).

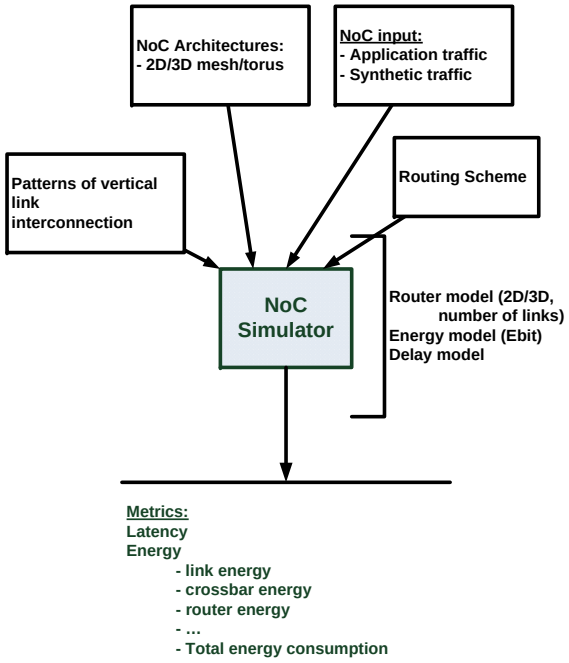
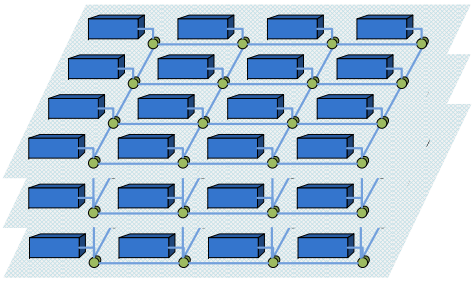
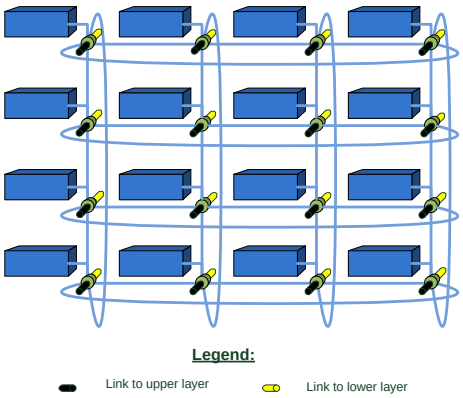


Figure 2: An overview of the exploration methodology of alternative topologies for application-specific 3D Networks-on-Chip.



(a) 3D Mesh.



(b) 3D Torus.

Figure 3: 3D NoC architectures: (a) Mesh and (b) Torus.

The NoC simulator is configured reading: a) the NoC architecture(two- or three-dimensional mesh and torus architectures, as well as defining the specific grid size (`x` and `y` parameters) and number of layers (`z` parameter), b) the type of input traffic, c) routing scheme, d) vertical link configuration file (which defines where vertical links are present or not) and e) the router model as well as the models used in order to calculate the energy and delay figures.

The output of the simulation is a log file containing the relevant cost factors we evaluate, such as: overall latency and average latency per packet, as well as the energy breakdown of the NoC, providing numbers for: link energy consumption, crossbar and router energy consumption etc. From these energy figures we calculate the total energy consumption of the 3D NoCs.

In order to support 3D architectures topologies, we have extended this simulator, providing additional routing schemes, and compatibility with the Trident traffic format [26]. Now the simulator supports 3D NoC architectures (3D Mesh and 3D Torus – as shown in Figure 3). The considered 3D architectures are composed of tiles that are connected using mesh and torus interconnection networks (are depicted in Figure 3). Each tile is composed of a processing core and a router. Since we are considering 3D architectures the router is connected to the six neighboring tiles and its local processing core via channels (consisting of two one-directional point-to-point links).

These 3D architectures to be explored may have a mix of two- and three-dimensional routers, from (very few 3D routers) to 100% (only 3D routers - 100% vertical interconnection link presence). In order to steer the exploration we are based on different patterns (as they were presented in Section 3. The proposed 3D NoCs can be constructed by placing a number of identical two-dimensional NoCs on individual layers, providing communication by inter-layer vias among vertically adjacent routers. This means that the position of silicon vias is exactly the same for each layer (this simplification is going to be tackled in future work). Hence, the router configuration is extended to the third dimension, while the structure of the individual logic blocks (IP cores) remains unchanged.

Furthermore, we have modified the routing procedure, shown in Algorithm 1 (valid for all routing schemes) in order to be able to route packets over the heterogeneous vertical link topologies. This modification allows the customization of the routing scheme in order to efficiently cope with the heterogeneous topologies (based on vertical link connectivity patterns).

The steps of the routing algorithm are:

1. For each packet we know the source and destination nodes and we can find the positions of these nodes in the topology. The positions of the nodes are: for the destination node `dst.x, dst.y, dst.z` and for the source one `src.x, src.y, src.z`.

2. By doing so we can formulate the *temporary destinations*, that is *one* temporary destination per layer. More specifically, for the number of layers a packet has to traverse in order to arrive to its final destination, the algorithm sets the route to a temporary destination located at position `dst.x, dst.y, src.z` initially. The algorithm takes under consideration the "direction" the packet is going to follow across the layers (i.e., if it is going to an upper or lower layer) and

Algorithm 1 Routing algorithm modifications

```
1: function RoutingXYZ
2:     src : Source Node
3:     dst : Destination Node
4:
5:     findCoordinates(); ▷ returns src.x, src.y, src.z, dst.x, dst.y and dst.z
6:
7:     for all layer ∈ NoC do
8:         if packet passes from layer then
9:             findTmpDestination();
10:        end if
11:    end for
12:    while tmpDestination NOT destination do
13:        packet ← tmpDestination;
14:    end while
15: end function

16: function findTmpDestination ▷ for each layer that the packet is going to traverse
17:    tmpDestination.x ← dst.x
18:    tmpDestination.y ← dst.y
19:    tmpDestination.z ← src.z
20:
21:    for all validNodes ∈ layer do
22:        if link NOT valid then ▷ information obtained through the vertical interconnections patterns input file.
23:            newLink ← computeManhattanDistance(); ▷ returns link with the smallest Manhattan distance
24:            tmpDestination ← newLink;
25:        else
26:            tmpDestination ← link;
27:        end if
28:    end for
29: end function
```

finds the nearest valid link at each layer, while updating properly the `z` coefficient of the temporary destination's position.

3. After finding a set of temporary destinations (each one located at a different layer), they are stored into the header flit of the packet.

4. The aforementioned temporary destinations may or may not be used, as the packet is being routed during the simulation, so they are "candidate" temporary destinations. The criterion of being just a candidate or the actual destination per layer is specified according to a set of vertical links that exhibited relatively high utilization during a previous simulation with the same network parameters and setting the desired minimum link communication volume or according to a given vertical link pattern as they are presented at Section 1.

5. A link is uniquely identified by the node that is connected to and its direction. So, for all the specified valid nodes that are located at the same layer with the header flit of the packet check if it matches with the desired for the route to the destination up/down link.

6. If there is no match between them, compute the Manhattan distance (in the case of 3D torus topology we have modified it in order to produce the correct Manhattan distance between the two nodes).

7. Finally, the valid link with the smallest Manhattan distance is chosen and its corresponding node is chosen to be the temporary destination at each layer the packet is going to traverse.

5. EXPERIMENTAL RESULTS

The main objective of the methodology and the exploration process is to find alternative 3D NoC topologies with a mix of 2D and 3D routers, exhibiting vertical link interconnection patterns that perform best to the incoming traffic. Our primary cost function is the energy consumption, with the other cost factors being the average packet latency and total switch block area. The comparison basis are the fully vertically interconnected 3D NoCs as well the 2D ones (with the same number of nodes).

The 3D router uses as a switching fabric a 7×7 crossbar switch, whereas the 2D one uses as a switching fabric a 5×5 crossbar switch. Additionally, each router has a routing table and based on the source/destination address, the routing table decides which output link the packet should deliver to.

As an energy model the NoC simulator is using the Ebit model, proposed in [27]. We make the assumption (based on the work presented in [23]) that the vertical communication links between the layers are electrically equivalent to horizontal routing tracks with the same length. In this way we consider that the energy consumption of a vertical link between two routers is the same one as the consumption of a link between two neighboring routers of the same layer.

Using the extended version of the Worm_Sim simulator, we have performed simulations involving a 64-node and a 144-node architecture with 3D mesh and torus topologies with synthetic traffic patterns (uniform, transposal and hotspot). The configuration files describing the corresponding link patterns are supplied to the simulator as an input. The size of the 3D NoCs we simulated were $4 \times 4 \times 4$ and $6 \times 6 \times 4$, whereas the 2D ones were the 8×8 and 12×12. We have used three types of input (synthetic traffic):

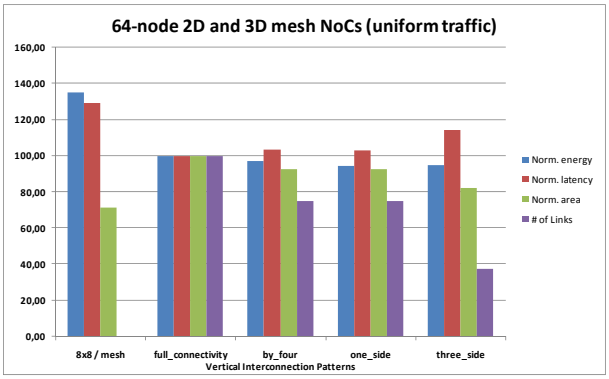
Uniform: Where we have uniform distribution of the traffic across the 3D NoC with the nodes receiving approximately the same number of packets.

Transpose: In this traffic scheme packets originating from node `x, y, z` have as destination the node (X - z, Y - y, Z - z), where `X, Y, Z` are the dimensions of the NoC.

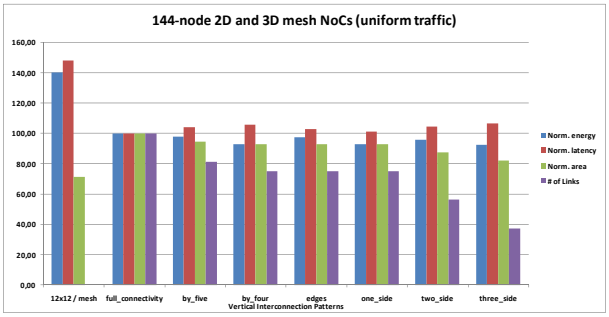
Hotspot: Where some nodes (a minority) receive increased number of packets (in our case it was at least 100% increased) than the majority of the rest of the nodes (which they receive packets in a uniform manner). The hotspot nodes in the 2D grids are positioned in the middle of every quadrant, where the size of the quadrant is specified by the dimensions of each layer in the 3D NoC architecture under simulation. Whereas, in the 3D NoC, a hotspot is located in the middle of each layer.

In Figure 4 the results of employing a non-fully vertical link connectivity to 3D mesh networks by using uniform traffic are presented. All the simulation measurements were taken for the same number of packets traversing the network. We can make a comparison of the total energy consumption, average packet latency and percentage of 2D routers (having 5 I/O ports instead of 7) under $4 \times 4 \times 4$ (Figure 4(a)) and $6 \times 6 \times 4$ (Figure 4(b)) mesh architecture. In the x-axis are presented all the interconnection patterns that exhibit lower energy consumption from the 3D NoC where all the routers are 3D (*full_connectivity*). The fact that not all patterns are

present in this Figure shows that not all of them can cope with this type of traffic. In the y-axis are presented, in a normalized manner – used as basis the figures of the full vertically interconnected 3D NoC, the cost factors for total energy consumption, average packet latency, total switching block area and number of links.



(a) Experimental results for a $4 \times 4 \times 4$ 3D Mesh.



(b) Experimental results for a $6 \times 6 \times 4$ 3D Mesh.

Figure 4: Uniform traffic on a 3D NoC for alternative interconnection topologies.

Employing the *by_four*, *one_side* and *three_side* link pattern results in up to 5% reduction in energy consumption, where 25% of the routers are designed to be 2D. As it can be seen, as we move to NoCs with greater dimensions, the same patterns exhibit better results. In 6x6x4 mesh3D architecture, the *by_five*, *by_four*, *edges*, *one_side*, *two_side*, *three_side* give up to 8% reduction in total energy consumption of the network, while 69.4% of vertical links are present. In the case of hotspot traffic, testing the $4 \times 4 \times 4$ mesh3D architecture, the 7 out of 9 link patterns perform better relatively to fully vertical connected topology. For instance, the *two_side* pattern exhibits 12% decrease in network energy consumption whereas the increase in latency is 2.5 cycles, note that only 56.25% of the vertical links are present. The hotspot traffic in 3D mesh topologies favors of cube topologies (for example $6 \times 6 \times 6$), even so, in $6 \times 6 \times 4$ mesh architecture the center and *two_side* exhibit similar performance compared to that of fully vertical connected architecture (that was expected due to the location where the hotspot nodes were positioned). Under the transpose traffic scheme, when the *by_four* link pattern is adopted it shows 6.5% decrease in total network energy's consumption at the expense of 3 cycles increased latency.

These results are also compared to their equivalent 2D architectures. For the 8x8 2D NoC (same number of cores to the 4x4x4 architecture) mesh architecture shows 25% increased latency and 40% increased energy compared to one side link pattern, whereas the 12x12 (same number of cores to the 6x6x4 architecture) mesh shows 46% increase in latency and 49% increase in energy consumption compared to the same pattern using uniform traffic. In addition, comparing the *by_four* pattern on 64-nodes architecture under transpose traffic shows 31% and 18% reduced latency and total network consumption, respectively. Whereas ,in case of hotspot traffic and employing the *two_side* link pattern, these numbers change to 24% reduced latency and 56% reduced energy consumption.

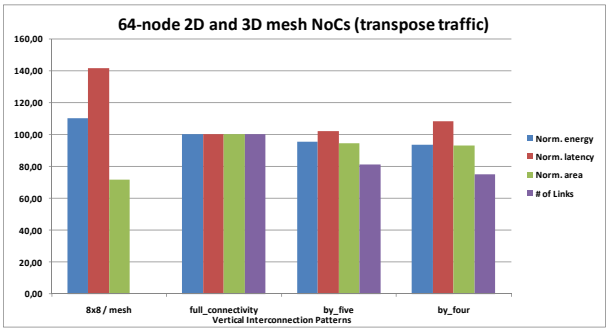
We also compared the performance of the proposed approach against that achievable with a torus network, which provides wrap around links added in a systematic manner. Note that the vertical links connecting the bottom with the upper layers are not removed, as this is the additional feature of the torus topology when compared to the mesh. Our simulations show that using the transpose traffic scheme, the vertical link patterns exhibit notable results, and this is goes better and better as the dimensions of the NoC get bigger. The explanation is that the flow of packets between a source and a destination is following a diagonal course among the nodes at each layer and this is also true the source-destination pair in 3D topologies, and this is where the wrap around links of the torus topology play a significant role in non reducing the performance even we remove some vertical links. And the results show that the bigger the dimensions of the NoC are, the energy savings also get bigger when the link patterns are applied. But, this is not true for the case of mesh topology. In particular, in the $6 \times 6 \times 4$ 3D torus architecture, using the *by_five*, *by_four*, *by_three*, *one_side*, *two_side* patterns show better results as long as the energy consumption is concerned, for instance, the *two_side* exhibit 7.5% energy savings and increased latency 32.84 cycles relatively to the 30 cycles of the fully vertical connected 3D torus topology.

In Figure 5 the simulation results for the 3D 4x4x4 mesh and 6x6x4 torus NoCs are presented for transpose type of traffic. From the Figure 5(a) we can see that we have a 4% gain in the energy consumption of the 3D NoCs with a 5% increase in the packet latency. Additionally we gain 6% in the area occupied by the switching blocks of the NoC (the numbers are derived from [8]). Comparing these patterns to the 2D NoC (having the same number of nodes) we can have on average a 14% decrease in energy consumption, a 33% decrease in total packet latency. But, on the area the cost of the 3D NoC is higher by 23%.

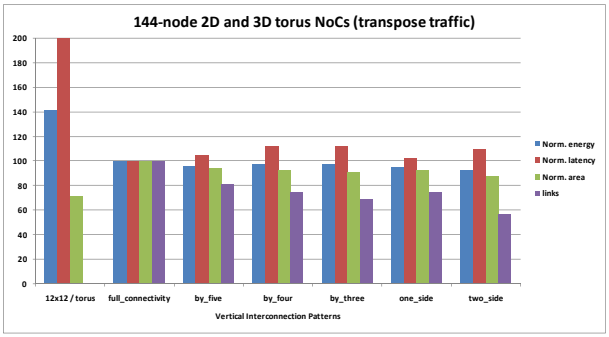
From the Figure 5(b) we can see that the 2D NoC experiences traffic contention and not being able to cope with that amount of traffic (the actual value of the latency is close to 5000 cycles per packet). Additionally, 47% gains achieved in energy consumption. When this torus architecture is compared to the "full" 3D one, it shows 5% gains in energy consumption with 8% increase in the latency and 9% reduces switching block area.

In Figure 6 the simulation results for the two 3D NoC architectures when triggered by a hotspot-type traffic are presented. In Figure 6(a) the results for the mesh architecture and in Figure6(b) the results for the torus architecture are presented respectively, showing gains in energy consumption and area, with a negligible penalty in latency.

What it can be seen from studying the analytical re-

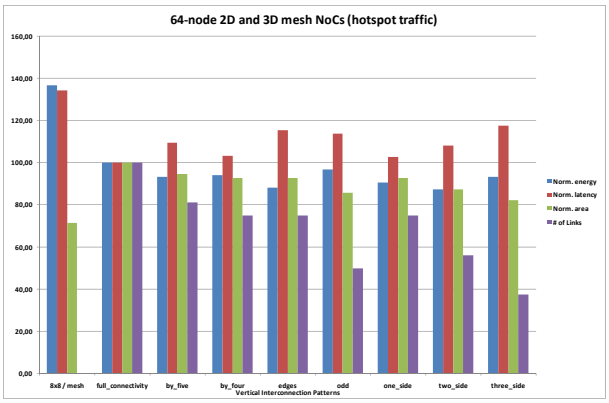


(a) Experimental results for a $4 \times 4 \times 4$ 3D Mesh.

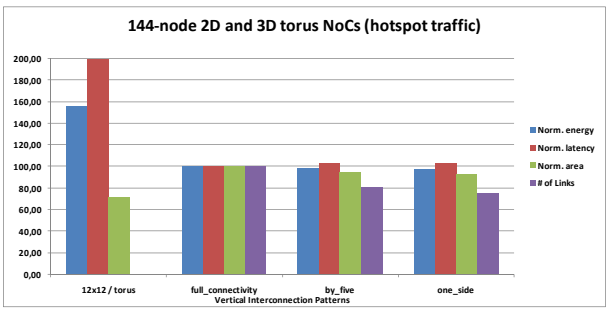


(b) Experimental results for a $6 \times 6 \times 4$ 3D Torus.

Figure 5: Transpose traffic on a 3D NoC for alternative interconnection topologies.



(a) Experimental results for a $4 \times 4 \times 4$ 3D Mesh.



(b) Experimental results for a $6 \times 6 \times 4$ 3D Torus.

Figure 6: Hotspot traffic on a 3D NoC for alternative interconnection topologies.

sults derived from Ebit [27] energy model, is that the link's, crossbar's, arbiter's, buffer's read energy consumption gets smaller in exchange with an increase in the energy consumed when writing to the buffer and by the router's routing engine. The results about the link and crossbar energy consumption for the three kinds of synthetic traffic used are shown in the Table below:

Table 1: Link and Crossbar energy consumption gains for all the used types of traffic.

		Traffic	
Energy Consumption	Uniform	Transpose	Hotspot
link	11% – 26%	7% – 36%	9% – 49%
crossbar	12% – 27%	7% – 38%	9% – 50%

Finally, in Table 2 the energy, latency and area values that were obtained are compared to the ones of the 3D full vertically interconnected NoC. The three types of traffic are shown in the 1st column. The next three columns present the gains (min. values to max. values – in %) for energy dissipation and area occupation (2nd and 4th column respectively). In the 3rd column the negligible increase in latency is presented.

Table 2: min-max impact on costs (energy, latency and area).

64-node architecture	Energy	Latency	Area
Uniform	98.9% – 94.6%	102.9% – 114.3%	92.9% – 82.2%
Transpose	98.9% – 92.9%	102.2% – 108.4%	94.7% – 92.9%
Hotspot	96.9% – 87.5%	102.8% – 117.6%	94.7% – 82.2%
144-node architecture	Energy	Latency	Area
Uniform	97.8% – 92.4%	101.3% – 106.7%	94.7% – 82.2%
Transpose	97.2% – 92.7%	102.1% – 112.3%	94.6% – 87.5%
Hotspot	99% – 97.5%	103.3% – 103.7%	94.7% – 92.9%

6. CONCLUSIONS AND FUTURE WORK

Networks-on-Chip are becoming more and more popular as a solution able to accommodate large numbers of IP cores, offering an efficient and scalable interconnection network. Three-dimensional NoCs are taking advantage of the progress of integration and packaging technologies offering advantages when compared to 2D ones. Existing 3D NoCs assume that every router of a grid can communicate directly with the neighboring routers of the same grid and with the ones of the adjacent layers. We have presented a methodology that shows that by employing an alternative 3D NoC vertical link interconnection network we can achieve gains in energy consumption (up to 12%) and in the area occupied by the routers of the NoC (up to 18%). The goal of the proposed methodology is to find heterogeneous 3D NoC topologies with a mix of 2D and 3D routers and vertical link interconnection patterns that performs best to the incoming traffic. In this way the exploration process evaluates the incoming traffic and the interconnection network, proposing an incoming traffic-specific alternative 3D NoC.

We want to further extend this work, not only exploring the 3D architectures but also the different router architectures, routing algorithms and performing further customizations targeting heterogeneous NoC architectures. In this way we are going to be able to create even more heterogeneous 3D NoCs. For triggering the NoCs we will move towards using real applications, apart from using even more types of synthetic traffic. By doing so, we will be able to propose application-domain-specific 3D NoCs architectures.

7. REFERENCES

[1] L. Benini and G. de Micheli. Networks on chips: a new soc paradigm. *Computer*, 35(1):70–78, 2002.

[2] E. Beyne. 3d system integration technologies. In *VLSI Technology, Systems, and Applications, 2006 International Symposium on*, pages 1–9, April 2006.

[3] E. Beyne. The rise of the 3rd dimension for system integration. In *Interconnect Technology Conference, 2006 International*, pages 1–5, June 5-7, 2006.

[4] E. Bolotin, I. Cidon, R. Ginosar, and A. Kolodny. Cost considerations in network on chip. *Integr. VLSI J.*, 38(1):19–42, 2004.

[5] M. Dall'Osso, G. Biccari, L. Giovannini, D. Bertozzi, and L. Benini. xpipes: a latency insensitive parameterized network-on-chip architecture for multi-processor socs. In *Proc. of ICCD*, page 536. IEEE Computer Society, 2003.

[6] W. Dally and B. Towles. *Principles and Practices of Interconnection Networks*. Morgan Kaufmann Publishers Inc., 2003.

[7] J. Duato, S. Yalamanchili, and N. Lionel. *Interconnection Networks: An Engineering Approach*. Morgan Kaufmann Publishers Inc., 2002.

[8] B. Feero and P. P. Pande. Performance evaluation for three-dimensional networks-on-chip. In *Proc. of ISVLSI*, pages 305–310, 2007.

[9] K. Goossens, J. Dielissen, and A. Radulescu. Æthereal network on chip: Concepts, architectures, and implementations. *IEEE Des. Test*, 22(5):414–421, 2005.

[10] W. Heirman, J. Dambre, and J. V. Campenhout. Synthetic traffic generation as a tool for dynamic interconnect evaluation. In *Proc. of SLIP*, pages 65–72. ACM Press, 2007.

[11] A. Jantsch and H. Tenhunen, editors. *Networks on chip*. Kluwer Academic Publishers, 2003.

[12] J. Joyner, R. Venkatesan, P. Zarkesh-Ha, J. Davis, and J. Meindl. Impact of three-dimensional architectures on interconnects in gigascale integration. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 9(6):922–928, Dec. 2001.

[13] J. Kim, C. Nicopoulos, D. Park, R. Das, Y. Xie, V. Narayanan, M. S. Yousif, and C. R. Das. A novel dimensionally-decomposed router for on-chip communication in 3d architectures. In *Proc. of ISCA*, pages 138–149. ACM Press, 2007.

[14] F. Li, C. Nicopoulos, T. Richardson, Y. Xie, V. Narayanan, and M. Kandemir. Design and management of 3d chip multiprocessors using network-in-memory. In *Proc. of ISCA*, pages 130–141. IEEE Computer Society, 2006.

[15] Z. Lu, R. Thid, M. Millberg, E. Nilsson, and A. Jantsch. Nnse: Nostrum network-on-chip simulation environment. In *Proc. of SSoCC*, April 2005.

[16] R. Marculescu, U. Y. Ogras, and N. H. Zamora. Computation and communication refinement for multiprocessor soc design: A system-level perspective. In *Proc. of DAC*, pages 564–592. ACM Press, 2004.

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

[18] U. Ogras and R. Marculescu. Application-specific network-on-chip architecture customization via long-range link insertion. In *Proc. of ICCAD*, pages 246–253, 6-10 Nov. 2005.

[19] U. Y. Ogras, J. Hu, and R. Marculescu. Key research problems in noc design: a holistic perspective. In *Proc. of CODES+ISSS*, pages 69–74, 2005.

[20] P. P. Pande, C. Grecu, M. Jones, A. Ivanov, and R. Saleh. Performance evaluation and design trade-offs for network-on-chip interconnect architectures. *Computers, IEEE Transactions on*, 54(8):1025–1040, Aug. 2005.

[21] V. Pavlidis and E. Friedman. 3-d topologies for networks-on-chip. In *Proc. of IEEE International SOC Conference*, pages 285–288, 2006.

[22] V. Puente, J. Gregorio, and R. Beivide. SICOSYS: an integrated framework for studying interconnectionnetwork performance in multiprocessor systems. In *Parallel, Distributed and Network-based Processing, 2002. Proceedings. 10th Euromicro Workshop on*, pages 15–22, 2002.

[23] R. Reif, A. Fan, K.-N. Chen, and S. Das. Fabrication technologies for three-dimensional integrated circuits. In *Quality Electronic Design, 2002. Proceedings. International Symposium on*, pages 33–37, 18-21 March 2002.

[24] F. Ridruejo and J. Miguel-Alonso. INSEE: An interconnection network simulation and evaluation environment. In *Proc. of Euro-Par Parallel Processing*, volume 3648/2005, pages 1014–1023. Springer Berlin / Heidelberg, 2005.

[25] V. Soteriou, N. Eisley, H. Wang, B. Li, and L.-S. Peh. Polaris: A system-level roadmap for on-chip interconnection networks. In *Proc. of ICCD*, October 2006.

[26] V. Soteriou, H. Wang, and L.-S. Peh. A statistical traffic model for on-chip interconnection networks. In *Proc. of MASCOTS*, pages 104–116. IEEE Computer Society, 2006.

[27] T. Ye, L. Benini, and G. De Micheli. Analysis of power consumption on switch fabrics in network routers. In *Proc. of DAC*, pages 524–529, 10-14 June 2002.