Performance and Energy Aware Inhomogeneous 3D Networks-on-Chip Architecture Generation

Michael Opoku Agyeman, Ali Ahmadinia *Member, IEEE,*, Nader Bagherzadeh *Fellow, IEEE,*

Abstract—Recently, Through-Silicon-Via (TSV) has been more popular to provide faster inter-layer communication in three-dimensional Networks-on-Chip (3D NoCs). However, the area overhead of TSVs reduces wafer utilization and yield which impact design of 3D architectures using a large number of TSVs such as homogeneous 3D NoCs topologies. Also, 3D routers require more memory and thus they are more power hungry than conventional 2D routers. Alternatively, hybrid 3D NoCs combine both the area and performance benefits of 2D and 3D router architectures by using a limited number of TSVs. Existing hybrid architectures suffer from higher packet delays as they do not consider the dynamic communication patterns of different application and their NoC resource usage. We propose a novel algorithm to systematically generate hybrid 3D NoC topologies for a given application such that the vertical connections are minimized while the NoC performance is not sacrificed. The proposed algorithm analyses the target application and generates hybrid architectures by efficiently redistributing the vertical links and buffer spaces based on their utilizations. Furthermore, the algorithm has been evaluated with synthetic and various real-world traffic patterns. Experimental results show that the proposed algorithm generates optimized architectures with lower energy consumption and a significant reduction in packet delay compared to the existing solutions.

Index Terms—Network-on-Chip, Multi-Processor System, 3D Integration, Performance Evaluation

F

1 Introduction

Three-dimensional Networks-on-Chip (3D NoC) has emerged as a promising technology that maintains the benefits of miniaturization by enabling higher integration density and enhancing system performance while providing a scalable communication platform for multi-core architectures [1].

Conventionally, homogeneous 3D NoCs have been employed for 3D-Integration where Through Silicon Vias (TSVs) have been used for interlayer communication [2]. Moreover, TSV manufacturing is an expensive and complicated process with high defect rates which causes poor yields [3], [4], [5]. Consequently, the homogeneous router distribution may lead to significant area overhead if applied to applications whose communication pattern vary significantly among embedded cores. To optimize the performance and manufacturing cost with minimal distortion to the modularity of 3D NoCs, hybrid architectures have been investigated to combine 2D and 3D routers in 3D NoCs [6]. Previous work on hybrid 3D NoCs has focused on different NoC router architectures, routing algorithm, minimal hop-count between 2D and 3D routers in each layer, and uniform distribution of 2D and 3D routers [6], [7], [8]. However, such hybrid combination in 3D NoC has not been exploited effectively, especially when we consider the vertical link utilization and buffer utilization of the target application. It might be argued that employing adaptive routing strategies would maximize bandwidth utilization. However this does not avoid underutilization in low traffic rates and also requires modification of the router architecture which may incur additional resources.

The main aim of this paper is to present a cost effective approach which systematically generates high performance 3D NoC architectures with minimum area and energy consumption by using limited number of vertical links based on link utilization along with distributing buffer spaces based on buffer utilization of the target application without introducing extra buffer or control resources. Additionally, this paper aims to facilitate design exploration of 3D NoC architectures with reduced number of TSVs to achieve optimized trade-off in terms of energy consumption and performance. Consequently, we present an evaluation of TSV variation based on the link utilization of target application to automatically generate low energy-high performance 3D NoC architectures. The paper is organized as follows: In Section 2, we present existing topologies and methodologies used for 3D NoC architectures. Section 3 introduces 3D NoC architecture characteristics as well as routing and mapping issues in inhomogeneous 3D NoCs. Section 4 presents a systematic approach to 3D NoC architecture generation. Section 5 evaluates the performance of generated architectures. Finally, the main findings are concluded in Section 6.

2 Related Work

Future Systems-on-Chips (SoCs) will have more heterogeneous cores and components which will require tailored NoC topologies [9], [10], [11]. Furthermore, 3D routers in 3D NoCs require more interconnections and arbitration compared to 2D routers. With increase in number of ports, the crossbar power consumption and occupied area increases

M. O. Agyeman is with the Intel Embedded System Research group, Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong
A. Ahmadinia is with the School of Engineering and Built Environment, Glasgow Caledonian University, Glasgow, UK
N. Bagherzadeh is with Dept. of Electrical Engineering and Computer Science, University of California, Irvine, Irvine, CA 92697, USA

significantly when 7 port symmetric 3D routers are used to replace conventional 2D routers [12]. Li et al proposed to replace the large 7 port symmetric 3D routers with 6 port NoC-Bus hybrid 3D routers introducing the Hybrid 3D NoC-Bus mesh architecture [13]. This architecture requires an addition arbiter per pillar to allow seamless integration of the Bus and NoC interface. Also, Xiangyu et al [14] have demonstrated that the area overhead of TSV increases with increase in number of 3D layers. Particularly, the area overhead of the TSV for a 4 layer 3D NoC with 5M gates can reach as high as 10%.

A study of the area and energy benefits of combining 2D and 3D routers in 3D mesh and torus topologies is presented in [8]. In their work, the placement of 3D routers does not consider the traffic load of the selected tile. These architectures will not perform well under high traffic conditions in different traffic patterns. Xu et al. [15] evaluated the impact of reducing the number of TSVs to half and quarter on the performance of 3D NoCs. Their proposed architectures, quarter/lo and half/lo (quarter/hi and half/hi), aim at generating 3D NoC with 2D routers placed as close to (far from) 3D routers as possible in each layer. These architectures suffer from uneven distribution of 3D routers and unpredictable delays for different applications. Similarly, 3D NoC architectures proposed in [6] do not consider the communication dynamics of applications.

Wang et al. [16] used partition islands of routers to constitute regions for sharing the same TSV pad for interlayer communication controlled by serialization logic. However, serialization along the TSV bundle causes the average packet delay to increase exponentially as the number of routers per TSV bundle increases. Also genetic algorithm and simulation annealing employed in [16] for the placement of different TSV patterns in 3D NoCs have an exponential complexity with a large design exploration space. Mishra et al. [17] proposed a heterogeneous 2D NoC architecture, where some routers ports have more virtual channels with wider interconnect links than others. This approach involves repacketization at different NoC regions to enable routing along different link widths. Such routers suffer from large area and power consumption. Kumar et al. [18] proposed a buffer-sizing algorithm for NoCs, where the buffers are increased iteratively based on the NoC's behavior under simulation. However, due to the iterative approach, the runtime of the simulation increases significantly as the number of nodes in the NoC increases.

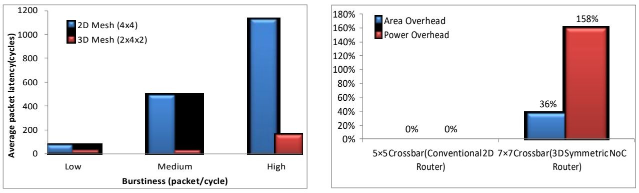
We propose a systematic approach to generate efficient 3D NoC architectures that anticipate performance of target applications by placing 3D routers at highly utilized vertical links. Moreover, the proposed method judiciously assigns more buffer resources to highly utilized channels in the routers while reducing that of lowly utilized ones without the use of virtual channel or the need for alteration of the routing algorithm. Unlike existing approaches, we generate simulation-driven optimized 3D NoC architectures with limited number of vertical links and optimized utilization of buffer spaces. Experiments conducted with both realistic and synthetic traffic patterns have demonstrated significant improvements in the performance when the proposed approach is applied to 3D NoCs with similar amount of resources (memory, 3D routers and links). Particularly, the consistent performance improvements obtained by applying a wide range of applications demonstrates that the proposed approach gives superior performance under varied applications though TSVs and Buffer space are statically redistributed and allocated.

3 3D NoC Architectures

Conventionally, 3D NoCs have smaller footprints and superior average packet latency compared to 2D NoCs even for an application with a small number of cores such as the Telecom benchmark [19], as shown in Figure 1(a). Contrary, Figure 1(b) shows that crossbars of 3D routers have much higher area and power consumption compared to that of 2D routers. We explore architectures that maximize the tradeoff benefits between 2D and 3D routers in NoCs. Most real world applications such as the Telecom benchmark have unbalanced traffic with different bandwidth requirements among inter-router links [21]. As a result, some links and buffers are highly utilized while others remain underutilized or redundant in such applications. This Section demonstrates that such utilization patterns even exist under uniform traffic patterns. Heterogeneous architectures tend to optimize the performance and manufacturing cost of 3D NoCs by using a limited number of TSVs. Existing hybrid 3D NoC architectures such as half/lo, quarter/lo, chess, xdiagonal and periphery enhance the performance of 3D NoCs by minimizing the average hop-count while evenly distributing 3D routers [6], [15]. However, we present an alternative approach to 3D NoC architecture generation that exploits the uneven resource utilization in NoCs.

3.1 Vertical Link Utilization

First, we investigate the link utilization of homogeneous 3D NoC under uniform traffic. Figure 2 shows that even under uniform traffic pattern where traffic is evenly distributed in 3D NoCs, some NoC resources are used more often than others. As can be seen in Figure 2(a), link utilization is higher at the center of the NoC than its corners. This is due to the nature of XYZ routing; central routers relay most of the traffic from different corners to their destinations. Additionally, Figure 2(b) demonstrates a non-uniform vertical link usage despite the equal distribution of vertical links and uniformity of the traffic pattern. One might argue that utilizations of the links in Figures 2(a) and 2(b) should be



(a) Average packet latency under Telecom benchmark

(b) Area and power overheads of 2D and 3D routers [20]

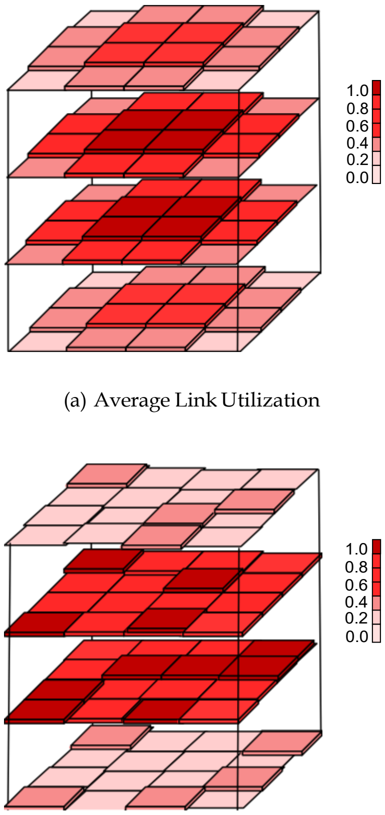
Fig. 1. Comparison of 2D and 3D mesh.

similar. However in XYZ routing, most of the traffic is first routed in the same layer (X and Y directions), and then along the vertical links between the source and destination layers. Also, experimental analysis presented in [22] confirms that data transmission in 3D NoC exhibits a temporal characteristic that neighboring vertical links are rarely busy at the same time.

Placing TSVs at links that are highly utilized in homogeneous architecture mimics the original traffic pattern and causes less distortion to the NoC traffic in the TSV limited architecture. Hence, there is less risk of congestion under deterministic routing in such architectures as packets experience fewer changes in their paths and take shorter routes to their destinations. As confirmed in Figures 7 and 8, *center* inhomogeneous architecture which has TSVs placed at the central nodes have higher average packet latency and energy consumption compared to architectures with TSVs placed at highly utilized vertical links.

3.2 Buffer Utilization

In order to exploit resources more efficiently, buffer utilizations of all router channels in the 3D NoC have been analyzed. Figure 3 demonstrates that congestion is more likely in the center of the mesh where there is more traffic in homogeneous 3D mesh under XYZ routing. Also, routers in the corners of the mesh have low utilization while the center routers are highly utilized and the remaining peripheral routers have intermediate buffer utilization. Similarly, exchange of traffic between top and bottom layers cause high buffer utilizations in the middle layers. However, the pattern of resource utilization differs for different NoC topologies and traffic patterns [17].



(a) Average Link Utilization

(b) Average Vertical Link Utilization

Fig. 2. Normalized utilization across all layers in a 4 → 4 → 4 3D Mesh

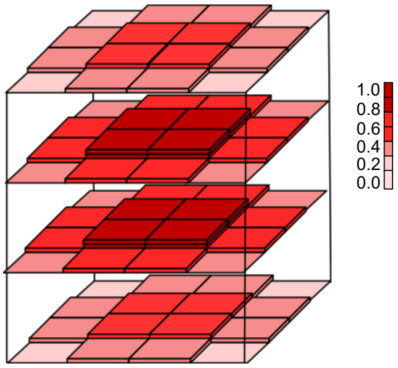


Fig. 3. Normalized buffer utilization across all layers in homogeneous 3D mesh under uniform traffic pattern.

In wormhole routing, packets have to wait in the input buffer of the router if the corresponding output buffer is full. Under high traffic conditions, the waiting time of packets can be longer which may cause the input buffers to be full. These conditions lead to hot spot regions and congestion in the network. One way of resolving this is to use more complex routing algorithms to reroute packets based on the available buffer resources. However, this requires extra routing logic and lead to high area overheads. Also, our analyses show that some routers have higher buffer utilizations than others even under uniform traffic pattern. Moreover, according to [23], buffers consume over 50% of the dynamic power consumption of the routers. An efficient redistribution of buffers in 3D NoCs will further increase the performance of NoCs without introducing virtual channels or repacketization while maintaining the same amount of buffer space.

3.3 Problem Formulation

In order to design a systematic flow for generating optimized 3D NoC architectures that considers the traffic flow and the employed routing strategy, we must first define the parameters [24] used in our proposed approach.

Definition 1. *Vertical Link Utilization (VLU) is a direct measure of workload on a vertical link i in a homogeneous 3D mesh for a simulation sampling period T which is defined as:*

$$VLU = \frac{\sum_{t=1}^{T} A(t)}{T}, 0 \le VLU \le 1 \tag{1}$$

where,

$$\vec{A}(t) = \begin{cases} 1 & \textit{if traffic passes } i \textit{ in cycle } t \\ 0 & \textit{if no traffic passes } i \textit{ in cycle } t \end{cases} \tag{2}$$

Definition 2. *Buffer utilization (BU) of a given port p is defined as:*

$$BU = \frac{\sum_{t=1}^{T} (\frac{b(t)}{B})}{T}, 0 \le BU \le 1 \tag{3}$$

where, b(t) is buffer space occupied at time t and B is the total buffer space.

Hence for a given router R_{xyz}, the total buffer utilization, BU_R in a given simulation sampling period T is given by:

$$BU_R = \sum_{i=1}^{P} BU_i \tag{4}$$

where, P is the total number of ports available at router R_{xyz} and BU_i is the buffer utilization of port i.

Definition 3. *A path allocation table (PAT) is a routing table for each node in a searching tree, which records paths of the application traffic among its occupied tiles.*

Definition 4. *Priority queue (PQ) of a* core_n *arranges all other cores in the 3D NoC in their ascending order based on a product of their bit energy*[1] *and Manhattan distance from* core_n *considering their routing path defined in their PAT relative to* core_n*.*

PAT ensures the cores with minimum cost, which is a direct measure of hop-count, relative to the node under consideration are arranged first. In order to generate an optimized 3D NoC architecture with uneven buffer distribution and a total of N_{3D} 3D routers for a given application, we exploit the average *VLU* across all layers to generate 3D NoC architecture with limited vertical link pillars. Our next goal is to enhance the performance of such architectures while reducing the area occupied by the buffer spaces by using routers with uneven buffer spaces based on the BU_R of the routers. To achieve this, it is necessary to implement efficient routing algorithms for packet path flow in hybrid 3D NoCs. Additionally, a 3D NoC mapping tool must be used to efficiently assign cores of real world applications.

3.4 Routing

One of the main contribution factors of NoC performance is the routing algorithm used to control traffic paths. We investigate our proposed methodology under both static and adaptive routing algorithms. Static routing uses a simple dimension-order deterministic routing algorithm. The determined route is a fixed route regardless of the dynamic nature of the network. Thus, all packets generated at source node and destined for a particular node at a different layer are forwarded to a predetermined 3D router. For intra-layer routing, we adopt simple XY routing between source and destination nodes. However, if source and destination nodes are in different layers, packets are forwarded along a 3D router that provides the shortest Manhattan distance between the source and destination nodes. Various adaptive routing techniques for TSV limited 3D NoCs have been proposed and proved to be deadlock and livelock free [7]. In addition, we perform our analysis under an efficient deadlock free adaptive routing algorithm (Buff_NVH) [7].

3.5 Performance and Energy Aware Mapping

To investigate the performance of 3D NoC architectures generated by the proposed approach in realistic applications, it is necessary to utilize an efficient mapping tool. Branch-and-bound has been extensively used as an optimal mapping algorithm for most studies in 2D NoCs [25], [26]. For a given NoC architecture, the algorithm efficiently steps through a searching tree which serves as a representation of the solution space while using heuristics to trim candidates that do not meet the desired design constraints. Consequently, we have extended the branch-and-bound technique for energy-aware mapping in 3D-NoCs with regular tile sizes which have a more complex searching tree compared to the 2D NoCs. Figure 4 shows an example of a searching tree for the 2x2x2 3D NoC (shown on the top right corner of the figure). For simplicity, we assume that nodes 6 and 7 are left unoccupied leaving a total of 6 cores to be mapped.

At the top of the tree is the root node which represents the state where no core has been mapped. Whereas a leaf node represents a completely mapped core and an internal node represents a partial mapping. A Path Allocation Table (PAT) is assigned to each node to record the routing path for communication traffic among its occupied tiles. The mapping algorithm automatically generates a PAT for each node. Also the PAT of each parent node is automatically inherited by each newly generated child node. The newly mapped nodes are assigned a routing path and added to the existing PAT. To ensure deadlock freedom and compatibility of the algorithm in hybrid architectures, routing algorithms described in Section 3.4 are employed to generate PAT for each node. Algorithm 1 describes the branch and bound algorithm used for energy and performance aware mapping of cores in 3D NoCs. The initialization phase arranges the cores by their communication volume in their descending order. Next the core with the highest inbound and outbound communication volume is greedily mapped unto the first tile on the layer closest to the heat-sink. This helps to reduce the size of the search tree and also reduces the energy consumption in the mapped application. Thus, this stage selects an unmapped core, assigns it to a tile and remuneratively assigns the remaining unmapped cores. A Priority Queue (PQ) is used to arrange the cores to be branched based on their cost in ascending order. Stepping through PQ in the ascending order to generate child nodes for a core during the branching process will likely decrease the minimum upper bound cost (UBC) to help detect non-promising mapping easily. The UBC [25] is a value that is no less than the minimum cost of its legal descendant leaf node. However, a lower bound cost (LBC) of a core is

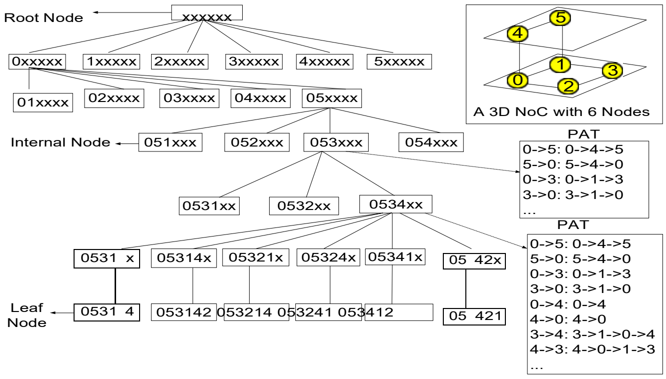


Fig. 4. Search tree for 3D NoC with 6 nodes

[1] Bit energy here refers to the total bit energy of the routers and links from core_n to the other core nodes.

Algorithm 1. Pseudocode of 3D mapping algorithm

```
Initialization Phase:
sort the cores by communication volume
root.node = new nodeNULL
MinumumUBC = + ┐ , bestMappingCost=+ ┐
PQ.Insertroot.node
Routing Phase:
while (!PQ.Empty()) do
  current.node = PQ.Next()
  for each unoccupied tile $t_x$ do
    generate a child node $core_{new}$
    allocate routing paths
    if $core_{new}$'s mirror node exists in the PQ then
      continue
    end if
    if $core_{new}$.LBC < MinumumUBC then
      if $core_{new}$.isLeafNode then
        if $core_{new}$.cost < bestmapping.cost then
          bestmapping.cost = $core_{new}$.cost
          bestmapping = $core_{new}$
        end if
      else
        if $core_{new}$.UBC < MinumumUBC then
          MinumumUBC = $core_{new}$.UBC
          PQ.insert(new)
        end if
      end if
    else
      break
    end if
  end for
end while
```

the lowest cost that its descendant leaf nodes can possibly achieve. Both the UBC and LBC are used in generating leaf nodes that give the best possible cost values. Here, the cost of a node to be mapped ($core_{new}.cost$) is the product of the Manhattan distance and energy consumed by the communication among the node and the tiles that have already been mapped. A core is rejected as a child node if its LBC is higher than the lowest UBC that has been found so far in the searching tree, since other leaf nodes will certainly lead to a better solution. If more than one leaf node is generated as a possible solution, the leaf node with the minimum energy consumption is selected as the best mapping solution. As Branch-and-Bound employs searching trees in finding solutions, the complexity of the mapping problem in Algorithm 1 increases exponentially with the number of variables [27], [28]. However, the proposed technique efficiently generates inhomogeneous 3D NoC architectures for a given mapped application.

4 Automatic Generation of Optimized 3D NoC Architectures

First, we propose efficient 3D NoCs with minimal TSV by considering the vertical link usage of the target traffic pattern. Secondly, we increase the performance of the architectures by introducing uneven buffer distribution in the architecture based on the buffer utilization of the target application under the TSV reduced generated architecture. Figure 5 simplifies the design flow of our proposed approach for optimized 3D NoC architecture generation. The application specifications and design constraints are taken as inputs. The communication data rate between embedded cores is specified in the application specification. For a realistic application, the application specification includes the cores and their associated tiles. Also 3D NoC size, total number of 3D and 2D routers expected in the final NoC architecture are taken as inputs. NoC models including 2D and 3D routers and links characterized with area, energy and frequency based on the target technology library are taken as inputs. The design flow involves two main stages:

Stage 1: With the specified parameters, a full system simulation is performed to generate the average vertical link utilization in homogeneous 3D NoC. This information is used to generate TSV reduced architectures where vertical links placed at highly utilized vertical links.
Stage 2: A further simulation is carried out with the newly generated NoC architecture to analyze the average buffer utilization in the router channels. An optimized 3D NoC architecture is then generated by resizing buffer spaces based on the buffer utilization.

The initial stage exploits the vertical link utilization of the given application in the simulated homogeneous 3D NoC to generate an energy efficient 3D NoC architecture. This is achieved by placing a designer specified number of 3D routers at the nodes with highly utilized vertical links. For a given x, y coordinate in a layer, the average vertical link utilization across all layers of the 3D NoC is collected. Hence, to maintain the regularity of the NoC while reducing the average hop-count, a vertical pillar is created by placing 3D routers at the same x, y coordinates in each layer for the nodes with high vertical link utilizations under the given constraint.

After stage 1, traffic redirection due to irregularity under either adaptive or static routing will cause further uneven buffer utilization. Thus leading to stage two, which evaluates the dynamics of buffer usage in the channels of the newly generated architecture and generates an optimized architecture. It should be noted that channels with more buffer spaces are generated by borrowing buffer spaces from other channels with low buffer utilization. Hence, in this paper, the total buffer space is kept constant without

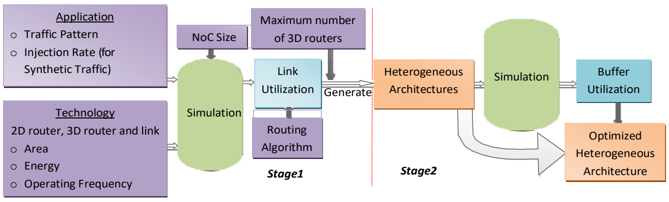


Fig. 5. Design flow for systematic generation of inhomogeneous 3D NoC architectures

sacrificing the performance of the NoC[2]. Also, a threshold determined by application profiling is used to determine the maximum and minimum buffer space allowed per channel. Without loss of generality, the minimum buffer at each port is determined by [18]:

$$|B_i| \geq 3 \times BU_i, \tag{5}$$

given that one hop requires only one clock cycle and $|B_i|$ is the buffer size of port *i*. The maximum buffer space is constrained by total available buffer space given by the user. The derived parameters are then employed to redistribute the buffer spaces as detailed in Algorithm 2. The algorithm takes the minimum and maximum allowed buffer sizes of port *i*, namely $|B_{min}|$ and $|B_{max}|$, respectively, as inputs. Additionally, BU_s which is a set of buffer utilizations of all the ports derived from the cycle accurate simulation analysis of the generated architecture is also taken as an input to the buffer sizing algorithm. The initialization phase arranges the ports in the network by their buffer utilization in the ascending and descending orders to create two lists: $List1$ and $List2$. Thus, $List1$ represents the ports with minimum buffer utilizations while $List2$ represents the ports with maximum buffer utilization. Next, buffer spaces are borrowed from each lowly utilized port (members of $List1$) and added to the highly utilized ports (members of $List2$). Here, we make sure that the resulting buffer size does not exceed the minimum ($|B_{min}|$) or maximum ($|B_{max}|$) allowed buffer space. This results in efficiently redistributed buffer spaces which have the same total number of buffers as the original 3D NoC. Consequently, in this study, we ensure that the total buffer space is not greater than that of a 3D NoC with uniform buffer distribution. It is obvious that some applications have unpredictable characteristics at run-time. However, our methodology focuses on most embedded applications, e.g. wireless and mobile applications which have regular communication patterns that can be profiled at design time [29]. Figure 6 shows an example of the placement of 3D routers in High25, High50 and High75 $4 \times 4 \times 4$ 3D NoC architectures generated from an application flow graph. High25, High50 and High75 represent architectures with a total of 25%, 50% and 75% 3D routers, respectively. In order to generate these architectures, initially the mapping algorithm is applied to map all the 61 cores to a $4 \times 4 \times 4$ 3D NoC, and then the vertical link utilization through simulation is obtained. Based on the link utilization values, the TSV locations in each architecture are determined. For an effective analysis of the performance benefits of our proposed methodology, we also generated Low25, Low50 and Low75 3D NoC architectures, where 3D routers are placed at the least utilized vertical links.

Algorithm 2. Pseudocode of buffer resizing algorithm

```
Initialization Phase:
List1 ← sort BU_s in ascending order
List2 ← sort BU_s in descending order
crnt1 = List1.first()
Buffer Resizing Phase:
while (!List1.Empty()) do
  for each value of m ∈ crnt1.buff_size do
    \\ m is set of integers {0...i.buff_size} of port i
    \\ i.buff_size is the total buffer size of port i
    if crnt1.buff_size − m > |B_min| then
      crnt2 = List2.first()
      while (!(List2.Index() = List2.last())) do
        if crnt2.buffer_size + m < |B_max| then
          crnt1.buff_size = crnt1.buff_size − m
          crnt2.buff_size = crnt2.buff_size + m
        else
          crnt2.buff_size = List2.Next()
        end if
      end while
    else
      crnt1.buff_size = List1.Next()
    end if
  end for
end while
```

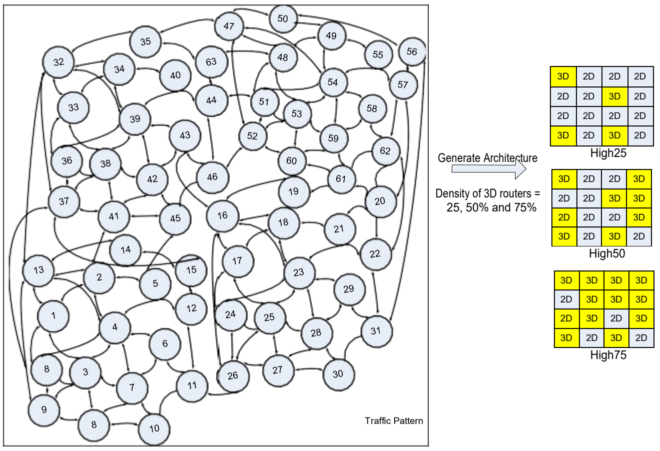


Fig. 6. Optimized architectures generation in 3D NoCs

In Algorithm 2, there are three main loops which must be considered in the worst case analysis of the proposed buffer resizing technique. The first *while* loop in the buffer resizing phase has two nested loops: *while* and *for* loops. The run time of the inner *while* and *for* loops depend on the number of ports and the buffer size of each port, respectively. Considering the worst case where all of the ports have relatively large buffer sizes in the network, the three loops have a complexity of $O(n^3)$. Thus within the outer *while* loop, the algorithm runs a constant number of instructions which is repeated *n* times (which is proportional to the ports). This instruction is repeated for a further *n* number of times within the nested *for* loop which is then repeated for additional *n* times. Also, Algorithm 2 has a partial dependency on the complexity of the sorting function which is $O(n \log_2 n)$. Therefore the complexity of the proposed resizing algorithm is $O(n^3)$. In addition to complexity analysis, the actual computation time of the whole design is also measured. The simulation framework is running in Windows 7 operating system on an i-7 (2.9 GHz) platform with 8GB RAM. For application sizes of 64 (NoC Size: $4 \times 4 \times 4$) and 32 (NoC Size: $2 \times 4 \times 4$) with uniform traffic patterns, the execution times of our simulation were

[2] This constraint could be eliminated to save more buffer space and power by slicing the buffer spaces at the expense of NoC performance

134 and 26 seconds respectively. These results correlate with the complexity of the proposed algorithms.

5 Evaluation

In order to evaluate the performance of the proposed architectures, a cycle-accurate NoC simulator is used by extending *Worm_sim* [26], an existing 2D NoC simulator. Our extended simulator accurately simulates 3D NoCs with any target configuration of 3D and 2D routers. We adopt parameters from [30], where the electrical characteristics of various NoC components for a $45nm$ CMOS process are presented. Energy consumption per bit of each component is then imported into the simulation platform and used to trace the energy profile of the entire NoC. Energy consumption of the NoC was estimated using E_{bit} energy model [31]. For energy analysis, both static and dynamic power of the router are calculated in Orion2.0 model for $45nm$ technology [32].

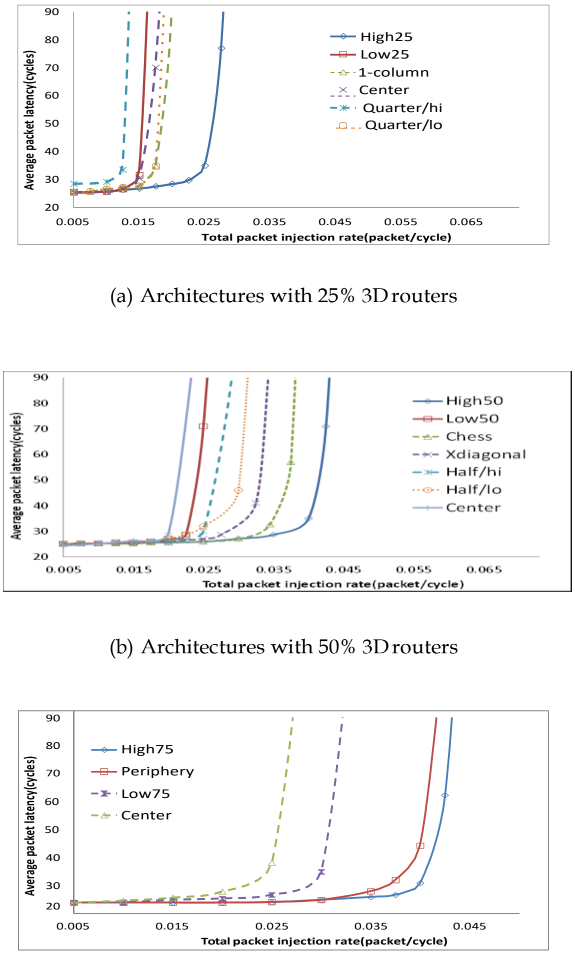
In order to evaluate the performance sustainability and traffic saturation points of the network, we performed our analysis under synthetic traffic patterns: uniform and hotspot. Moreover, to investigate the performance and energy of the NoC in real-world scenarios, realistic traffic patterns from different application domains have been considered: a complex multimedia traffic (MMS) [26], Auto-indust and Telecom (from the E3S benchmark) [19] and an AV (Audio-visual) benchmark [33]. Based on the size of the benchmark, NoC sizes of $4 \times 4 \times 4$ or $3 \times 4 \times 2$ have been used in the simulation. The setup is run for a warm-up period of 2000 cycles and performance statistics are collected after a simulation period of 200, 000 simulation cycles. Hence, by introducing different router models in the system, we have compared the average packet latency and energy consumption.

5.1 Experimental Results under Uniform Traffic Pattern

Under uniform random traffic pattern, each node has equal probability of communication with other nodes. Evaluating a NoC under this traffic pattern forecasts a worst case scenario of high inter-node communication density due to equal communication among nodes. We evaluate the performance of 3D NoC architectures under both static and adaptive routing for worst case analysis.[3]

5.1.1 Evaluation under Deterministic Routing

In this section, we first look at the latency and energy of the automatically generated 3D NoC architectures compared to the existing ones under static routing. Figure 7 shows that architectures with links placed at highly utilized vertical links have lower average packet delays compared to existing hop-count based hybrid architectures with equivalent number of 2D and 3D routers under uniform traffic pattern. As can be seen in Figure 7(a), High25 saturates with a much higher traffic load than other architectures. It can also be observed that, Low25, quarter/hi and center architectures have the highest average packet latencies. The architecture performance boundary becomes even more clearly defined as we increase the number of 3D routers as shown in Figures 7(b) and 7(c). Particularly, center architecture has much lower average packet latency compared to other TSV reduced architectures as can be seen in Figure 7(c). This is because in center architecture, TSVs are placed at the central nodes causing a redirection of interlayer traffic with a pattern similar to Figures 2(a) and 3 under static routing. However, these nodes have higher blocking probabilities, therefore causing packets to experience longer delays. Figure 8 summarizes the average packet energy of various 3D NoC architectures. It can be observed that, packets in the proposed architectures have the lowest energy compared to architectures with equivalent number of 3D routers. It can also be deduced that, packets in Low25, Low50 and Low75 architectures have high packet energies though they have limited number of TSVs. This is mainly due to the increased paths introduced by redirecting packets along vertical links with low utilization. On the other hand, the average packet energy in homogeneous architectures is similar to that of Low25 and Low50 architectures, because homogeneous ar-



(a) Architectures with 25% 3D routers

(b) Architectures with 50% 3D routers

(c) Architectures with 75% 3D routers

Fig. 7. Average packet latency under uniform traffic pattern and static routing.

[3] We compare our work with heterogeneous architectures (discussed in Section 2) proposed in [6] where 3D routers have fixed placement but evenly distributed in the NoC and [15] where the placement of 2D and 3D routers ensure minimum hop within the 2D layers.

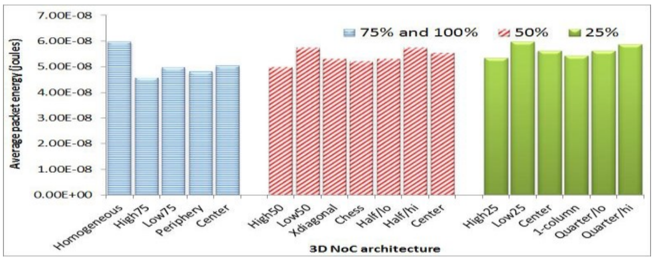


Fig. 8. Average packet energy under uniform traffic pattern and static routing. 100% represents homogeneous 3D NoC. 75%, 50% and 25% represent the total number of 3D routers present in the NoC architecture.

chitectures have extra switching activities with more 3D routers and interlayer connectivity.

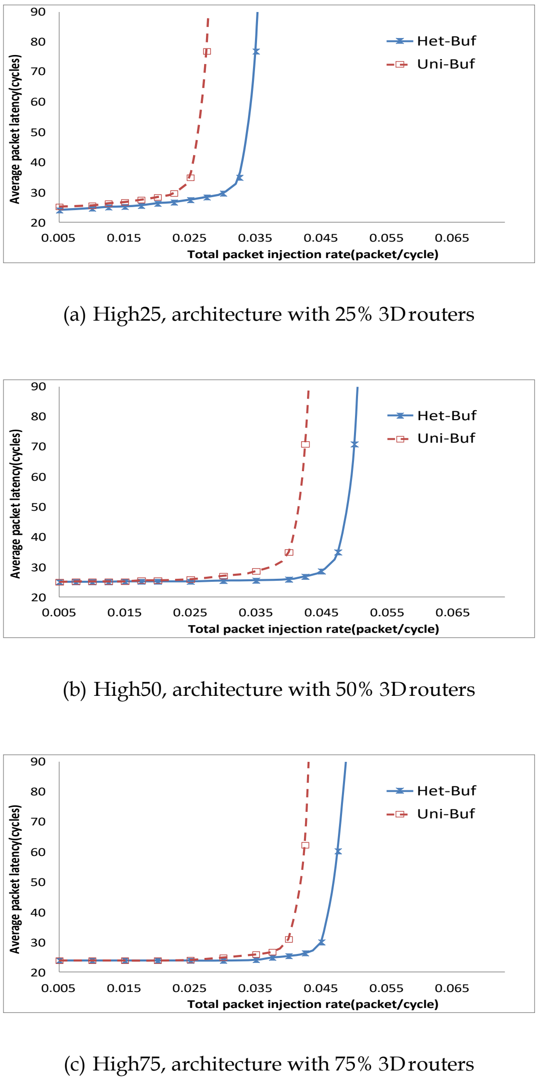
In summary, it can be observed from the simulation results that the proposed systematic approach generates architectures with smaller average hop count compared to existing hybrid architectures with equivalent number of 3D routers. Consequently, 3D NoC architectures generated by our proposed technique have lower average packet latency and energy compared to existing architectures.

5.1.2 Effect of buffer distribution under Static Routing

In this section, the impact of buffer redistribution on the proposed link utilization-based 3D NoCs is investigated. Figure 9 demonstrates that by introducing heterogeneous buffer distribution, the performance of our proposed architectures can further be improved despite their superiority in delay and energy over existing architectures. Most importantly, it can be noticed that architectures with uneven buffer distribution (Het-Buf) saturate with higher injection loads compared to architectures with uniform buffer distribution (Uni-Buf). This is expected as Het-Buf considers the congestion state of the buffers in the target application. Although adding more TSVs is expected to improve the performance, static routing does not fully utilize the path diversity in the NoC. Hence, the maximum improvement in the average packet latencies in Figures 9(b) and 9(c) are similar. However, it is evident that channels in architectures with the non-uniform buffer distribution have lower blocking probability and hence lower packet delays.

5.1.3 Evaluation under Adaptive Routing

A common method for improving average packet latency is adopting deadlock-free adaptive routing techniques. However, due to redirection of traffic in adaptive routing, the performance improvements vary for different architectures. Figure 10 illustrates a comparison of the average packet latency of architectures generated by our systematic approach and existing hop-count based 3D NoC architectures with equivalent number of 2D and 3D routers under adaptive routing. As can be seen, though there is a general increase in the NoC saturation load with adaptive routing, the performance of NoC architectures with links placed at highly utilized vertical links is better than other architectures with similar number of 3D routers. However, Figure 11 reveals that the average packet energy of the proposed architectures is lower than that of existing architectures due to shorter communication paths between nodes. Figure 12 demonstrates that, though the performance of architectures with links placed at highly utilized vertical links is better than that of existing architectures, we can further improve their performance to saturate with an even higher injection load when non-uniform buffer distributions are employed. Figures 12(a), 12(b) and 12(c) reveal the superiority of Het-Buf, non-uniform buffer redistribution generated by our systematic approach across architectures with different total number of 3D routers. By reassigning buffer spaces based on application traffic pattern in target application, adaptive routing exploits the non-uniformity of the architectures to improve the average packet latency of the NoC.



(a) High25, architecture with 25% 3D routers

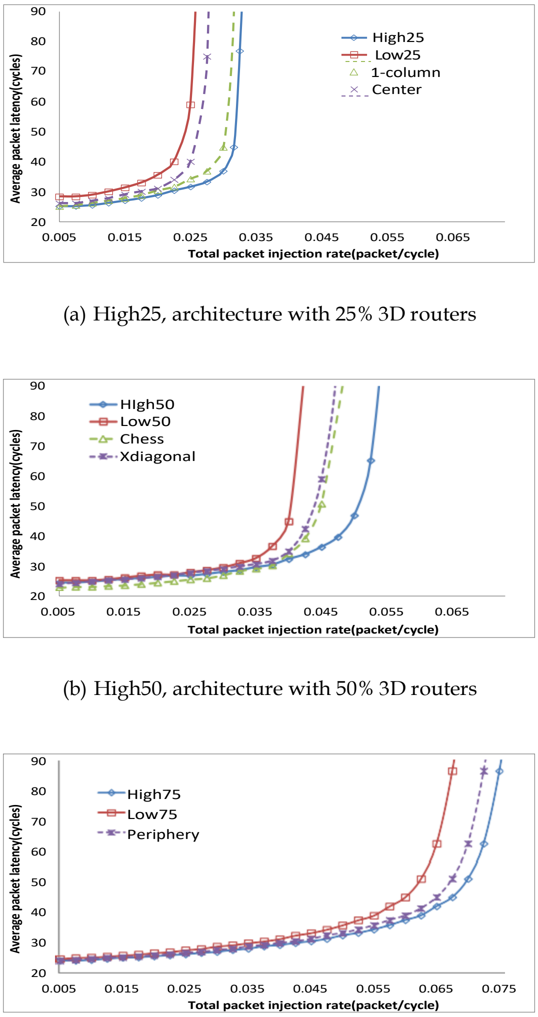
(b) High50, architecture with 50% 3D routers

(c) High75, architecture with 75% 3D routers

Fig. 9. Impact of buffer distribution on average packet latency (uniform traffic)

5.2 Experimental Results under Hotspot Traffic Pattern

In realistic applications, some nodes receive more packets than others causing different hotspot regions. Hotspot is a synthetic traffic pattern, which has some selected hotspot nodes receiving more traffic than others. The remaining traffic is sent uniformly to all other nodes. It should be noted



(a) High25, architecture with 25% 3D routers

(b) High50, architecture with 50% 3D routers

(c) High75, architecture with 75% 3D routers

Fig. 10. Average packet latency under uniform traffic and adaptive routing.

that, several hotspot scenarios were experimented to study the effectiveness of our method, while a selection of them due to page limits is presented. In all these experimented cases, our method improved the results. Here a center node in the middle layer is selected as the hotspot node with

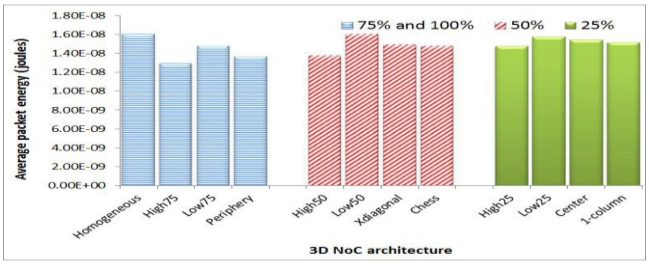
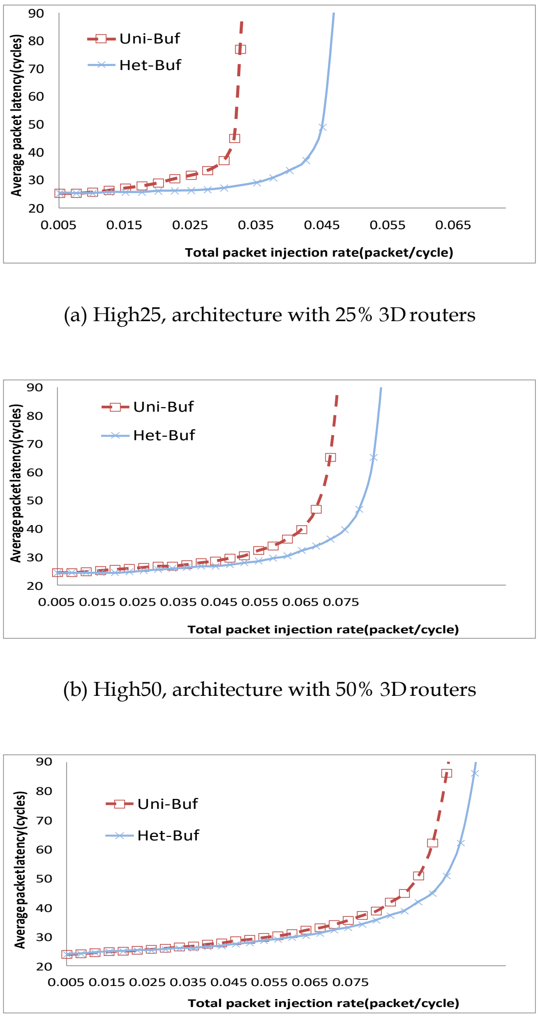


Fig. 11. Average packet energy under uniform traffic and adaptive routing. 100% represents homogeneous 3D NoC. 75%, 50% and 25% represent the total number of 3D routers present in the NoC architecture.



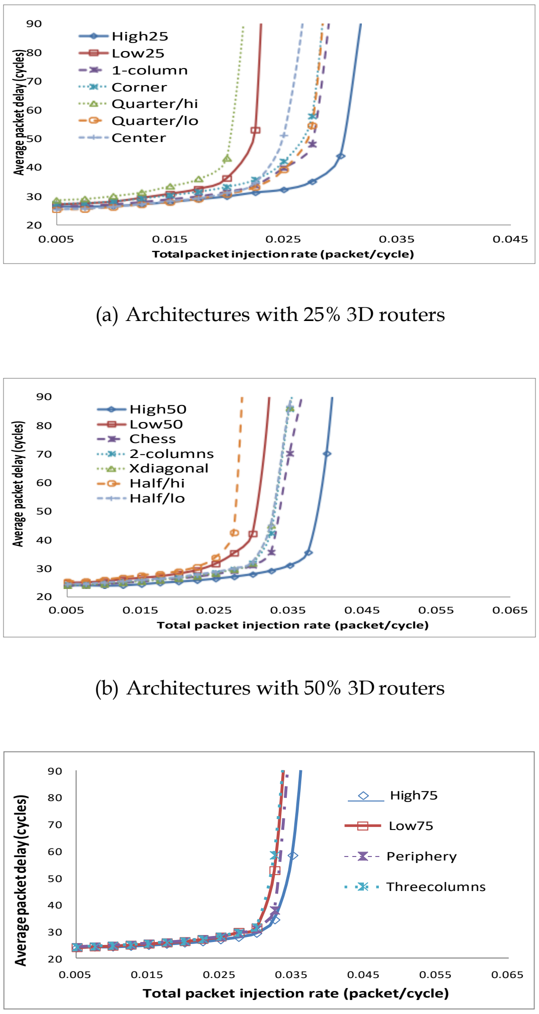
(a) High25, architecture with 25% 3D routers

(b) High50, architecture with 50% 3D routers

(c) High75, architecture with 75% 3D routers

Fig. 12. Impact of buffer distribution on average packet latency (uniform traffic)

10% higher chance of receiving packets than all other nodes. As can be seen in Figure 13, architectures with links placed at highly utilized vertical links have lower average packet latency than existing 3D NoCs with the same number of 3D routers. Moreover, Figure 14 shows that architectures generated by the proposed systematic approach, high25, high50 and high75 are not only superior in average packet latency but also in terms of average packet energy. It can be noticed that, for architectures with equivalent number of 3D routers and even in the case of homogeneous 3D mesh, architectures with links placed at highly utilized vertical links have lower average packet energy consumption. By increasing the number of hotspot nodes from 1 to 4, each with 10% higher chance of receiving packets than other nodes, Figure 15 shows the performance benefits of architectures generated by our systematic approach, especially in Figures 15(b) and 15(c), when the number of hotspots are increased to 3 and 4 nodes, respectively. It can be noticed that the performance of architectures with vertical links placed at



(a) Architectures with 25% 3D routers

(b) Architectures with 50% 3D routers

(c) Architectures with 75% 3D routers

Fig. 13. Average packet latency under hotspot traffic pattern

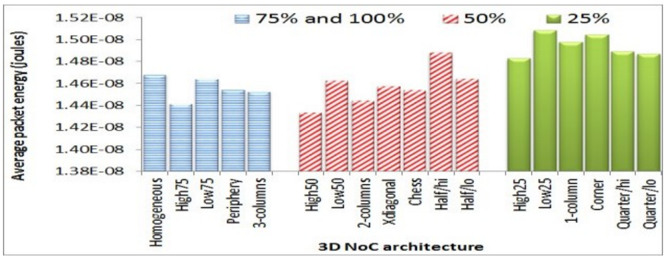
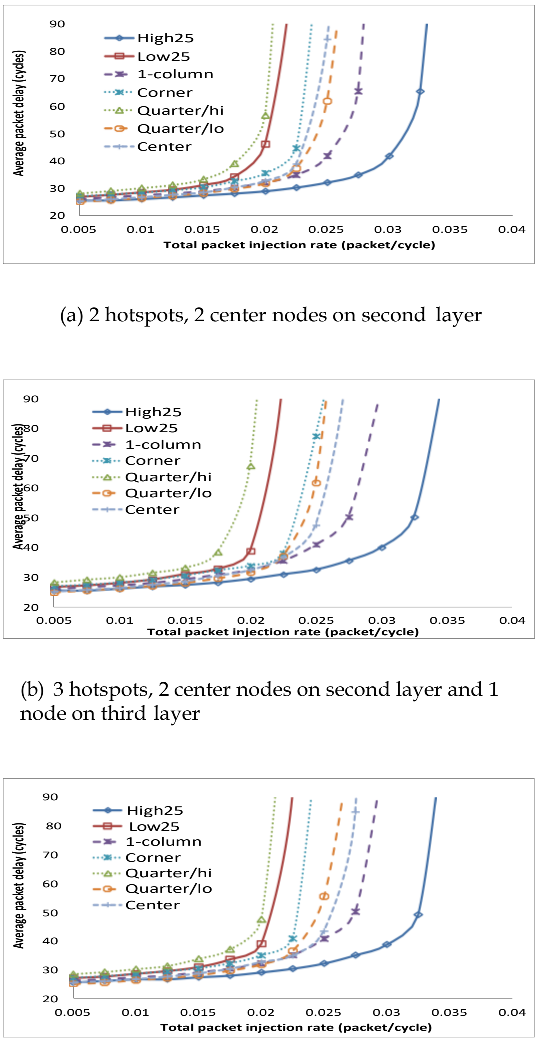


Fig. 14. Average packet energy under hotspot traffic pattern. 100% represents homogeneous 3D NoC. 75%, 50% and 25% represent the total number of 3D routers present in the NoC architecture.

highly utilized links increases as the number of hotspots increases while that of existing architectures decreases with increased packet delays causing the NoC to saturate with lower injection rates. This is expected as the proposed approach regenerates optimized architectures for each hotspot configuration by considering the vertical link utilization.



(a) 2 hotspots, 2 center nodes on second layer

(b) 3 hotspots, 2 center nodes on second layer and 1 node on third layer

(c) 4 hotspots, 2 center nodes on second and third layers

Fig. 15. Variation of average packet latency of 3D NoC architectures with various number of hotspot nodes

5.3 Experimental Results under Realistic Applications

In this section, the proposed approach is evaluated with a range of realistic traffic patterns from different application domains. For the MMS, Telecom, AV and Auto-indust case studies, we present performance results for NoC architectures (High) generated by our approach and compare them with existing architectures with equivalent number of vertical links and buffer resources. Figure 16 shows that the proposed 3D NoC architectures have lower average packet latencies in all cases. It can be noted that the average packet latencies of the proposed architectures outperform that of homogeneous 3D mesh though they have a limited number of vertical links.

Moreover, Figure 17 shows that the proposed High25, High50 and High75 architectures have lower average packet energy compared to architectures with equivalent number of 3D routers. On the other hand, high average packet

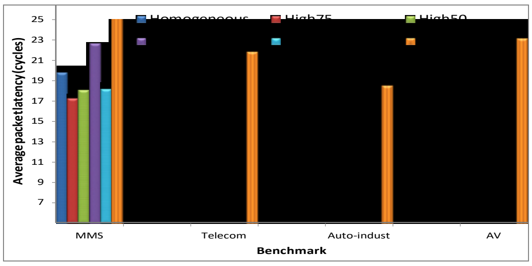


Fig. 16. Average packet latency of 3D NoC architectures under realistic traffic patterns

energy was recorded in homogeneous 3D mesh. It can be concluded that TSV reduced NoC architectures can be employed in realistic applications for higher performance than homogeneous 3D mesh and with even more efficient energy consumption.

To validate the performance gains of architectures generated by the proposed approach, we have applied a varied set of benchmarks: V OPD, D_26__media, D38__TV OPD, D36_4 and D64_4 [34]. A summary of the NoC configurations of the benchmarks is given in Table 1, where the number of cores and NoC dimensions are highlighted. Quarter/lo and half/lo architectures for these configurations were generated by placing the 2D routers as close to the 3D routers as possible in each layer as described in [15]. Similarly, Figure 18 shows that architectures generated by our systematic approach have superior performance compared to other architectures under realistic traffic patterns. Particularly, Figure 18(a) demonstrates that High25, High50 and High75 3D NoC architectures have lower average packet latencies than conventional homogeneous 3D mesh.

Benchmark	Number of Cores	NoC Dimensions
D_26_media	26	4 ⇥ 4 ⇥ 2
D_38__tvopd	38	4 ⇥ 4 ⇥ 3
D_62_pvopd	62	4 ⇥ 4 ⇥ 4
D_36_4	36	3 ⇥ 4 ⇥ 3
D_64_4	64	4 ⇥ 4 ⇥ 4

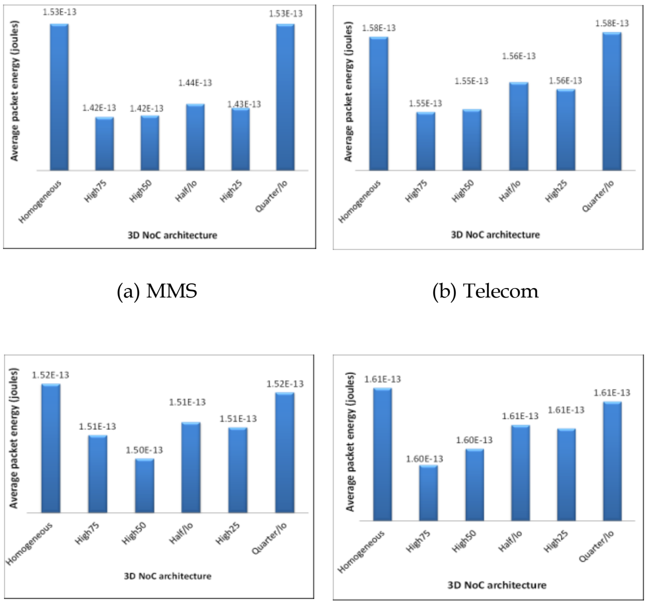
TABLE 1
Summary of Bechmarks

Moreover as shown in Figure 18(b), the proposed 3D NoC architectures have lower average packet energies compared to homogeneous 3D mesh in all cases.

5.4 Impact Analysis of TSV Variation

In this section, we explore TSV variation in highly utilized vertical links to identify optimal or near optimal trade-off points in terms of performance and power consumption. Here, the power consumption of the 3D NoC includes the power values for 2D and 3D routers, the physical links and the TSVs. Therefore, an increase in TSV bundle (as shown in Figure 19) also involves adding a 3D router.

As illustrated in Figure 19, the average packet latency of 3D NoCs decreases as the number of TSV pillars increases at the expense of power efficiency. This is because the connectivity in the NoC increases with higher switching activity as the number of TSVs increases. It can be seen that the normalized average packet latency of architectures with 6 to 15 TSV pillars (which is a total of 37.5%-93.7% 3D routers) is similar. In contrast, the power consumption is significantly high for architectures with high number of TSVs. Also, as shown in Figure 20, architectures with 7



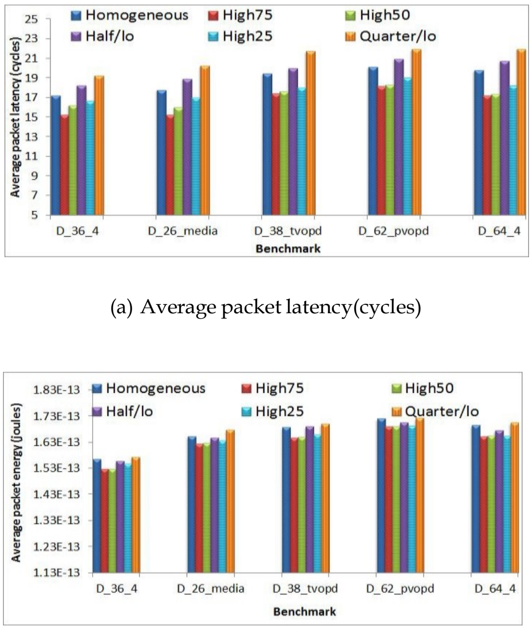
(a) MMS

(b) Telecom

(c) Auto-Indust

(d) AV

Fig. 17. Average packet energy under realistic traffic patterns



(a) Average packet latency(cycles)

(b) Average packet energy(joules)

Fig. 18. Performance under other realistic traffic patterns.

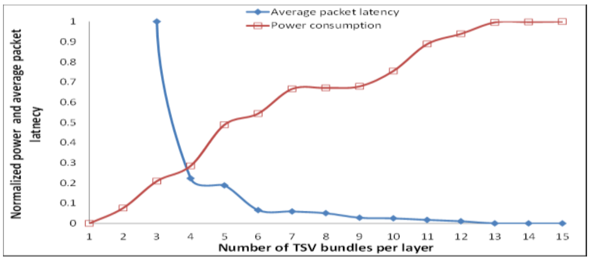


Fig. 19. Trade-off analysis of TSV pillars variation under uniform traffic pattern

TSV pillars (43.75% 3D routers) have the same maximum sustainable load as that of architectures with maximum number of TSVs (with 50 - 93.7% 3D routers). Contrarily, the maximum sustainable load for 3D NoC architectures with 6 TSV pillars before network saturation is significantly low under uniform traffic. Therefore, in applications with high probability of data communication among all nodes such as uniform traffic, placing a total of 43.75% TSV pillars at highly utilized vertical links provide near optimized power and performance efficiency, while saving 56.25% of area and power hungry 3D routers. In real-world applications, however, the inter-node communication is more defined with some nodes exchanging more data than others. As shown in Figure 21, power consumption of 3D NoCs in MMS increases rapidly when the number of TSV pillars is greater than 11. Also, the average packet latency is comparable when the number of TSV pillars is greater than 3. Hence under this real-world application, we can save up to a total of 75% TSVs without sacrificing the NoCperformance.

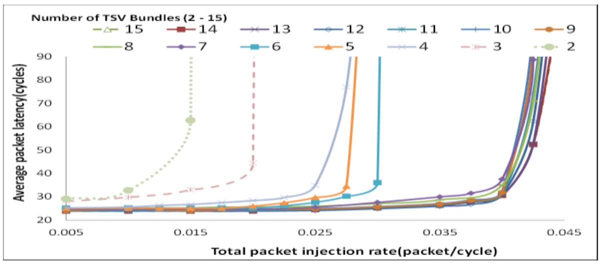


Fig. 20. Variation of TSV pillars with average packet latency under uniform traffic pattern.

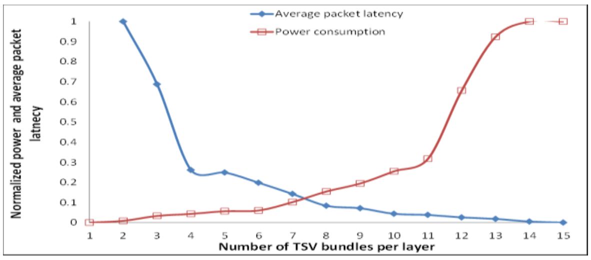


Fig. 21. Trade-off analysis of TSV pillars variation (MMS traffic).

5.5 Comparison with Existing Approaches

In our proposed approach, we exploit the buffer and link utilization to generate 3D NoCs with minimized number of TSVs. So far, we have compared our 3D NoCs with topologies that have the similar distribution of routers. Most application specific 3D NoCs, on the other hand, have varied topologies for different architectures. Similarly in [35], TSV count has been explored for a thermally aware 3D NoC floorplanner. This tool generates an initial 3D NoC architecture including placement of cores and routers for a given application. Routers are placed based on shortest paths between source and destination nodes. The resulting architecture is fed back into the floorplanner and this process is repeated until a predetermined cost function such as TSV count is satisfied. We have compared our approach in terms of mapping cores and TSV placement with [35]. VOPD and MPEG4 benchmarks from their results are selected, as we had used the same benchmarks in our evaluations. Both applications have 12 cores and network size of 3⇥ 2 ⇥ 2. In the floor planning process, 9 and 14 TSVs have been used for VOPD and MPEG4 respectively [35]. We have used the same technology, network size and number of TSVs in our 3D NoC generation. As can be seen in Table 2, the power results are similar, as the number of TSVs is equal in both approaches. However, our approach generated lower packet delay compared to the simulated allocation model due to more optimized placement of TSVs in the 3D NoC.

We have also compared 3D NoCs generated by our approach with two application specific techniques [21], [36] as similar benchmarks have been used in both cases. We have used the same data width of the NoC links (32 bits), NoC operating frequency (400 MHz), technology (65 *nm*), TSV model [37] and maximum number of inter layer links (25 links) in our 3D NoC generation. Figure 22 shows the power consumption of 3D NoC architectures generated by our proposed technique and the application specific 3D NoCs: 3D Sunfloor [21] and 3D layer-by-layer [36]. As these two papers only provide their power results, we could not compare other metrics such as latency. As shown in Figure 22, 3D layer-by-layer has a much higher power consumption compared to 3D NoCs generated by our approach and 3D Sunfloor [21]. Both 3D Sunfloor and 3D layer-by-layer try to find optimized link lengths between router-to-router connection as well as router-to-PE connection. However, 3D layer-by-layer connects PEs in a layer to routers in the same layer. Consequently, packets need to travel through more routers to reach the destination nodes and hence resulting in a higher power consumption compared to 3D Sunfloor [21]. Though packets need to be rerouted in architectures generated by the proposed approach, the power consumption is similar to 3D Sunfloor. This is because, 3D NoCs generated

Benchmark	VOPD Power (mW)	MPEG4 Power (mW)	VOPD Delay (ns)	MPEG4 Delay (ns)
[35]	81	103	5.1	6.3
proposed	78.1	101.2	4.5	5.7

TABLE 2
Comparison with Simulated Allocation technique [35]

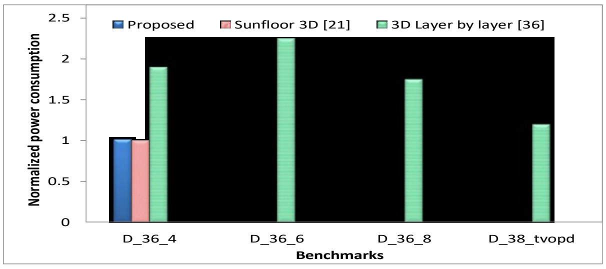


Fig. 22. Comparison with 3D Sunfloor [21] and layer-by-layer [36]

by 3D Sunfloor have several local ports per router which increase the work load of the routers and hence increases the dynamic power consumption for buffering, arbitration and router traversal.

In [16], various 3D NoC architectures are evaluated based on TSV serialization (Se-2 and Se-4), TSV sharing (TS-4 and TS-2), TSV sharing with planar adaptive routing (TS2-PA), and concentrated mesh with high radix routers (C4Mesh and C2Mesh). For comparison with our proposed approach, we have evaluated High50 and High25 with XYZ and adaptive routing algorithms using the same simulation settings as presented in [16] for a 2 →4→ 4 NoC size. Tables 3 and 4 compare the normalised (based on a homogeneous 3D NoC) saturation points of different architectures with 25% and 50% 3D routers, respectively. As can be seen, architectures with adaptive routing perform better than XYZ routing ones. Overall, our optimized architectures provide higher saturation points for all cases (whith the same routing algorithm) under uniform traffic pattern. For shuffle traffic, the difference in adaptive routing cases is small, but with static routing, our architectures achieve higher saturation points. Again, this proves that consideration of link and buffer utilizations can improve the latency and performance in 3D NoC architectures.

Architecture	Uniform Traffic	Shuffle Traffic
TS4	0.7	0.89
Se-4	0.6	0.44
C4mesh	0.36	0.4
High25 (adaptive)	0.82	0.98
High25	0.78	0.93

TABLE 3
Comparison of normalised latency saturation points with TSV economizing architectures (25% 3D router) [16]

Architecture	Uniform Traffic	Shuffle Traffic
TS2-PA	0.96	0.98
TS2	0.9	0.96
Se-2	0.88	0.76
C2mesh	0.48	0.58
High50 (adaptive)	0.98	0.98
High50	0.94	0.98

TABLE 4
Comparison of normalised latency saturation points with TSV economizing architectures (50% 3D router) [16]

6 Conclusion and Future Work

In this paper, we have proposed an efficient and systematic approach to generate optimized 3D NoC architectures based on the resource utilization in the target application. Both the proposed and existing NoC architectures with different configurations of 2D and 3D routers in 3D mesh topology have been modelled and evaluated with a cycle accurate simulator under synthetic and realistic traffic patterns. Our evaluation under the widely used uniform and hotspot traffic patterns demonstrates that optimized 3D architectures generated by our systematic approach have lower average packet latency, lower energy consumption and can sustain more network load compared to existing architectures with equivalent number of 2D and 3D routers. Especially in uniform traffic pattern, a performance improvement up to 60% in the average packet latency is observed when vertical links are placed on highly utilized vertical links for architectures with 25% 3D routers. Experimental analysis under realistic traffic patterns shows that, NoC architectures with 3D routers placed at nodes with highly utilized vertical links achieve superior performance compared to architectures with the same number of buffer resources and vertical links. In general, this performance is further improved with an average of 25% when non-uniform buffer distributions are employed. Evaluated TSV number variation in 3D NoCs have demonstrated that in applications with high inter-node data rates, we can save up to 56.25% of TSVs without sacrificing the average packet delay while increasing the energy efficiency. Moreover, in realistic applications, even with reductions of up to 75% in the number TSVs, generated 3D NoCs have similar performance as architectures with maximum number of TSVs.

The proposed systematic flow is simulation driven and hence there is a trade-off time with accuracy and performance improvement using this approach. Therefore, future work includes a queuing theory and/or network calculus based approach to generate performance and energy aware inhomogeneous 3D NoC architectures.

References

[1] B. Feero and P. Pande, “Networks-on-Chip in a Three-Dimensional Environment: A Performance Evaluation,” *IEEE Transactions on Computers*, vol. 58, no. 1, pp. 32 –45, 2009.

[2] “Tour guide to 3d-ic design tools and services,” http://www.gsaglobal.org/eda/docs/, accessed: 04/2013.

[3] C. Feng, M. Zhang, J. Li, J. Jiang, Z. Lu, and A. Jantsch, “A low-overhead fault-aware deflection routing algorithm for 3d network-on-chip,” in *IEEE Symposium on VLSI (ISVLSI)*, 2011, pp. 19–24.

[4] D. Velenis, M. Stucchi, E. Marinissen, B. Swinnen, and E. Beyne, “Impact of 3d design choices on manufacturing cost,” in *IEEE Conference on 3D System Integration (3DIC)*, 2009, pp. 1 – 5.

[5] X. Dong, J. Zhao, and Y. Xie, “Fabrication cost analysis and cost-aware design space exploration for 3-d ics,” *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 29, no. 12, pp. 1959–1972, 2010.

[6] M. O. Agyeman, A. Ahmadinia, and A. Shahrabi, “Heterogeneous 3d network-on-chip architectures: area and power aware design techniques,” *Journal of Circuits, Systems and Computers*, vol. 22, no. 4, p. 1350016, 2013.

[7] ——, “Efficient routing techniques in heterogeneous 3d networks-on-chip,” *Parallel Computing*, vol. 39, no. 9, pp. 389–407, 2013.

[8] K. Siozios, A. Bartzas, and D. Soudris, “Three dimensional network-on-chip architectures,” in *Networks-on-Chips: Theory and Practice*, F. Gebali, H. Elmiligi, and M. W. El-Kharashi, Eds. CRC Press, 2009, pp. 1–28.

[9] S. Borkar, "Thousand core chips: a technology perspective," in *Design Automation Conference (DAC)*, 2007, pp. 746–749.

[10] R. Marculescu, U. Ogras, L.-S. Peh, N. Jerger, and Y. Hoskote, "Outstanding research problems in noc design: System, microarchitecture, and circuit perspectives," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, no. 1, pp. 3 –21, 2009.

[11] C. Weis, I. Loi, L. Benini, and N. Wehn, "An energy efficient dram subsystem for 3d integrated socs," in *Design, Automation Test in Europe Conference Exhibition (DATE)*, 2012, pp. 1138 –1141.

[12] 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," *SIGARCH Comput. Archit. News*, vol. 35, no. 2, pp. 138–149, 2007.

[13] 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 *International Symposium on Computer Architecture (ISCA)*, 2006, pp. 130–141.

[14] Y. Xie, J. Cong, and S. Sapatneker, "System-level 3d ic cost analysis and design exploration," in *Three Dimensional Integrated Circuit Design*. Springer US, 2010, pp. 261–280.

[15] T. Xu, P. Liljeberg, and H. Tenhunen, "A study of Through Silicon Via impact to 3D Network-on-Chip design," in *Conference On Electronics and Information Engineering*, 2010, pp. 333 – 337.

[16] Y. Wang, Y.-H. Han, L. Zhang, B.-Z. Fu, C. Liu, H.-W. Li, and X. Li, "Economizing TSV Resources in 3D Network-on-Chip Design," *IEEE Transactions on Very Large Scale Integration Systems*, 2015.

[17] A. K. Mishra, N. Vijaykrishnan, and C. R. Das, "A case for heterogeneous on-chip interconnects for CMPs," in *Annual International Symposium on Computer Architecture (ISCA)*, 2011, pp. 389–400.

[18] A. S. Kumar, M. P. Kumar, S. Murali, V. Kamakoti, L. Benini, and G. D. Micheli, "A buffer-sizing algorithm for network-on-chips with multiple voltage-frequency islands," *J. Electrical and Computer Engineering*, vol. 2012, 2012.

[19] R. Dick, "Embedded system synthesis benchmarks suite (e3s)," {http://ziyang.eecs.umich.edu/dickrp/e3s/}, accessed: 10/2012.

[20] Y. Xie, N. Vijaykrishnan, and C. Das, "Three-dimensional network-on-chip architecture," in *Three Dimensional Integrated Circuit Design*, ser. Integrated Circuits and Systems, 2010, pp. 189–217.

[21] C. Seiculescu, S. Murali, L. Benini, and G. De Micheli, "Sunfloor 3d: A tool for networks on chip topology synthesis for 3-d systems on chips," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 29, no. 12, pp. 1987–2000, Dec 2010.

[22] C. Liu, L. Zhang, Y. Han, and X. Li, "Vertical interconnects squeezing in symmetric 3D mesh Network-on-Chip," in *Asia and South Pacific Design Automation Conference*, 2011, pp. 357 –362.

[23] S. Murali, T. Theocharides, N. Vijaykrishnan, M. Irwin, L. Benini, and G. De Micheli, "Analysis of error recovery schemes for networks on chips," *IEEE Design and Test of Computers*, vol. 22, pp. 434 – 442, 2005.

[24] L. Shang, L.-S. Peh, and N. Jha, "Dynamic voltage scaling with links for power optimization of interconnection networks," in *High-Performance Computer Architecture*, Feb 2003, pp. 91–102.

[25] C. Ababei, H. S. Kia, O. P. Yadav, and J. Hu, "Energy and reliability oriented mapping for regular networks-on-chip," in *Proceedings of Symposium on Networks-on-Chip (NOCS)*, 2011, pp. 121–128.

[26] J. Hu and R. Marculescu, "Energy- and performance-aware mapping for regular NoC architectures," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 4, pp. 551–562, 2005.

[27] P. K. Hamedani, S. Hessabi, H. Sarbazi-Azad, and N. E. Jerger, "Exploration of temperature constraints for thermal aware mapping of 3d networks on chip," in *International Conference on Parallel, Distributed and Network-based Processing*, 2012, pp. 499–506.

[28] N. Thakoor and J. Gao, "Branch-and-bound for model selection and its computational complexity," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, pp. 655–668, 2011.

[29] W. H. Ho and T. M. Pinkston, "A methodology for designing efficient on-chip interconnects on well-behaved communication patterns," in *International Symposium on High-Performance Computer Architecture (HPCA)*, Washington, DC, USA, 2003, pp. 377–.

[30] V. Pavlidis and E. Friedman, "3-D Topologies for Networks-on-Chip," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 15, no. 10, pp. 1081–1090, 2007.

[31] T. T. Ye, G. D. Micheli, and L. Benini, "Analysis of power consumption on switch fabrics in network routers," in *Proceedings of Design Automation Conference (DAC)*, 2002, pp. 524 – 529.

[32] A. Kahng, B. Li, L.-S. Peh, and K. Samadi, "Orion 2.0: A fast and accurate noc power and area model for early-stage design space exploration," in *Design, Automation Test in Europe Conference Exhibition (DATE)*, 2009, pp. 423 –428.

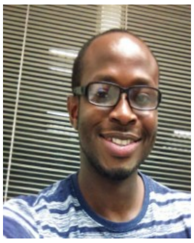
[33] V. Dumitriu and G. Khan, "Throughput-oriented noc topology generation and analysis for high performance socs," *VLSI*, vol. 17, no. 10, pp. 1433 –1446, 2009.

[34] C. Seiculescu, S. Murali, L. Benini, and G. De Micheli, "3d Network on Chip Topology Synthesis: Designing Custom Topologies for Chip Stacks," in *3D Integration for NoC-based SoC Architectures*. Springer New York, 2011, pp. 193–223.

[35] P. Zhou, P.-H. Yuh, and S. S. Sapatnekar, "Optimized 3d network-on-chip design using simulated allocation," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 17, no. 2, pp. 12:1–12:19, Apr. 2012.

[36] S. Murali, C. Seiculescu, L. Benini, and G. De Micheli, "Synthesis of networks on chips for 3D systems on chips," in *Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2009, pp. 242–247.

[37] I. Loi, F. Angiolini, and L. Benini, "Supporting vertical links for 3D networks-on-chip: toward an automated design and analysis flow," in *Proc. of intnl. conference on Nano-Networks*, 2007, pp. 1 – 5.



Michael Opoku Agyeman received the BSc. (Hons.) in electrical and electronics engineering from Kwame Nkrumah University of Science and Technology (KNUST), Ghana, in 2008, and the MSc. degree in embedded and distributed systems from London South Bank University, London, in 2009. He received the PhD from the department of computing at Glasgow Caledonian University, Glasgow, in 2014. Currently, he is with the Intel Embedded System Research group of The Chinese University of Hong Kong as a research Associate. His research interests include VLSI SoC design, reconfigurable computing, wired and wireless NoCs.

Ali Ahmadinia received his Ph.D. degree from University of Erlangen-Nuremberg, Germany, in 2006. In 2004-2005, he worked as a research associate in Electronic imaging group, Fraunhofer Institute - Integrated Circuits (IIS), Erlangen, Germany. In 2006-2008, he was a research fellow in the School of Engineering and Electronics, University of Edinburgh, Edinburgh, UK. In 2008, he joined Glasgow Caledonian University, Glasgow, UK, where he is now a senior lecturer in embedded systems. His research has resulted more than 80 international journal and conference publications and nearly 1 million pounds worth of grants in the areas of reconfigurable computing, system-on-chip design, wireless and DSP applications.

Nader Bagherzadeh (F'14) is a professor of computer engineering in the department of electrical engineering and computer science at the University of California, Irvine, where he served as a chair from 1998 to 2003. Dr Bagherzadeh has been involved in research and development in the areas of: computer architecture, reconfigurable computing, VLSI chip design, network-on-chip, 3D chips, sensor networks, and computer graphics since he received a Ph.D. degree from the University of Texas at Austin in 1987. He is a Fellow of the IEEE. Professor Bagherzadeh has published more than 250 articles in peer-reviewed journals and conferences. He has trained hundreds of students who have assumed key positions in software and computer systems design companies in the past twenty five years. He has been a PI or Co-PI on more than $8 million worth of research grants for developing next generation computer systems for applications in general purpose computing and digital signal processing.