

# Optimizing 3D NoC Design for Energy Efficiency: A Machine Learning Approach

Sourav Das, Janardhan Rao Doppa,  
Dae Hyun Kim, Partha Pratim Pande  
School of EECS, Washington State University  
Pullman, WA, USA  
Email: {sdas, jana, dkim2, pande}@eeecs.wsu.edu

Krishnendu Chakrabarty  
Department of ECE, Duke University  
Durham, NC, USA  
Email: krish@ee.duke.edu

**Abstract**— Three-dimensional (3D) Network-on-Chip (NoC) is an emerging technology that has the potential to achieve high performance with low power consumption for multicore chips. However, to fully realize their potential, we need to consider novel 3D NoC architectures. In this paper, inspired by the inherent advantages of small-world (SW) 2D NoCs, we explore the design space of SW network-based 3D NoC architectures. We leverage machine learning to intelligently explore the design space to optimize the placement of both planar and vertical communication links for energy efficiency. We demonstrate that the optimized 3D SW NoC designs perform significantly better than their 3D MESH counterparts. On an average, the 3D SW NoC shows 35% energy-delay-product (EDP) improvement over 3D MESH for the nine PARSEC and SPLASH2 benchmarks considered in this work. The highest performance improvement of 43% was achieved for RADIX. Interestingly, even after reducing the number of vertical links by 50%, the optimized 3D SW NoC performs 25% better than the fully connected 3D MESH, which is a strong indication of the effectiveness of our optimization methodology.

**Keywords**— *Small-World, 3D NoC, Discrete Optimization, Machine Learning.*

## I. INTRODUCTION

Three-dimensional (3D) ICs are capable of achieving better performance, functionality, and packaging density compared to the traditional planar ICs [1]. On the other hand, network-on-chip (NoC) enables integration of large numbers of embedded cores in a single die. 3D NoC architectures combine the benefits of these two new paradigms to offer an unprecedented performance gain [2]. With freedom in the third (vertical) dimension, NoC architectures that were previously impossible or prohibitive due to wiring constraints in planar ICs are now realizable in 3D NoC, and many 3D implementations can outperform their 2D counterparts. However, existing 3D NoC architectures predominantly follow straightforward extensions of regular 2D NoC designs, which do not fully exploit the advantages provided by the 3D integration technology [2].

In this paper, we consider the design space of 3D small-world (SW) NoC architectures, where the vertical connections mostly work as long-range shortcuts for SW networks. The key challenge is to place the long-range shortcuts optimally to achieve the desired goal. We formulate an objective function called communication cost and leverage machine learning to intelligently explore the combinatorial space of 3D SW NoC architectures to optimize this objective. Eventually, it helps us to achieve low latency and less energy consumption. We show that the proposed 3D SW NoC outperforms the state-of-the-art NoC architectures on multiple benchmarks. We also

demonstrate the efficacy and robustness of our optimization methodology by producing 3D SW NoC architectures that can perform as well as or better than the fully connected 3D MESH with significantly less number of vertical links.

The rest of the paper is organized as follows: Section 2 describes the related work. We present problem formulation, proposed solution, and the optimization algorithm in section 3. In Section 4, we present the experimental results and related analysis. Finally, Section 5 concludes the paper by summarizing the salient features of this work.

## II. RELATED PRIOR WORK

Most of the existing 3D NoC architectures utilize a conventional mesh [2][3][4]. However, it is well-known that mesh-based architectures suffer from high network latency and energy consumption due to its multi-hop communication links. To exploit the reduced distance along the vertical dimension of 3D IC, NoC-bus hybrid architecture was proposed in [5] that uses Dynamic Time Division Multiple Access (dTDM) to reduce the network latency. To reduce energy consumption of the system, the 3D Dimensionally Decomposed (DimDe) NoC router architecture [6] was developed. Reducing the number of input ports, an improved version of 3D NoC router architecture was developed in [7]. All of these architectures have buses in the Z-dimension; and hence, with increase in the network size, they are subject to traffic congestion and high latency under high traffic injection loads.

Despite recent advances in TSV technologies, TSVs are still subject to manufacturing defects and wearout [8], so researchers have developed NoCs with partial vertical connections [9]. To compensate for the loss in performance due to TSV failure, fault tolerant router and NoC architectures with redundant vertical links [10] were proposed. However, these designs give rise to additional area and power overheads.

The Sunfloor 3D was developed for synthesizing application specific 3D NoCs [11]. The design of application-specific 3D NoC architectures was also investigated in [12][13]. Later, more general-purpose 3D NoC was proposed in [14] using an ILP based algorithm to insert long-range links to develop low diameter and low radix architecture. However, the reduction in energy consumption was found to be limited.

Photonic interconnects offer high bandwidth and low power for future multi-core chip design. A number of hybrid 3D/photonic NoC architectures [15][16] were designed considering these benefits. However, on-chip photonics still suffer from performance variation due to thermal issues [17]. In addition, the challenges of integrating two emerging paradigms,

namely 3D IC and silicon nano-photonics, are yet to be adequately addressed.

In this work, we focus on designing a robust 3D NoC architecture that combines the benefits of 3D ICs and the robustness of the SW architecture. We present a detailed design methodology and a machine learning based optimization algorithm for developing energy-efficient 3D NoC architectures. We also perform comparative performance analysis with respect to conventional 3D MESH and other irregular architectures. Finally, exploiting the inherent robustness of SW network in the presence of link failure, we also show that our proposed 3D NoC outperforms traditional NoCs even in the presence of significant TSV failures without the need of any extra resources.

### III. OPTIMIZATION OF 3D NoC

In this section, we first describe our design problem and then present a high-level overview of the proposed machine learning based optimization methodology. Next, we provide the specific details of all the main components of our method for 3D NoC optimization.

**Problem Description:** The goal of on-chip communication system design is to transmit data with low latencies and high throughput using the least possible power and resources. In this context, the design of SW network based NoC architectures [18] is a notable example. It has been shown that either by inserting long-range shortcuts in a regular mesh architecture to induce a SW effect or by adopting power-law based SW connectivity, it is possible to achieve significant performance gain and lower energy consumption compared to the traditional multi-hop mesh networks [18][19]. In this work, we advocate that the concept of small-worldness should be adopted in 3D NoCs too. Specifically, the vertical links in 3D NoC should enable the design of long-range shortcuts necessary for a SW network. However, the appropriate placement of the planar and the long-range links along the vertical dimension are crucial for maximizing the performance benefits. Hence, our goal is to optimize the placement of the planar and vertical links in a 3D NoC where the overall interconnection architecture follows the power-law based connectivity [19]. The probability of having a direct link between nodes in a SW network varies exponentially with the link length, i.e.,  $p(\ell) \propto \ell^{-\alpha}$ . The parameter  $\alpha$  governs the nature of connectivity, e.g., a larger  $\alpha$  means a locally connected network with a few, or even no long-range links. By the same token, a zero value of  $\alpha$  generates an ideal SW network following the Watts-Strogatz model [20] – one with long-range shortcuts that are virtually independent of the distance between the cores.

**Small-world (SW) Network:** An SW network lies in between a regular, locally interconnected mesh network and a completely random Erdős-Rényi topology. SW graphs have a very short average path length, defined as the number of hops between any pair of nodes. The average shortest path length of SW graphs is bounded by a polynomial in  $\log(N)$ , where  $N$  refers to the number of nodes; this property makes SW graphs particularly interesting for efficient communication with minimal resource [19].

Starting from a power-law based connectivity, we attempt to optimize the location of the horizontal and vertical links to achieve lower latency and energy consumption. We define an objective function  $O$  called communication cost, which

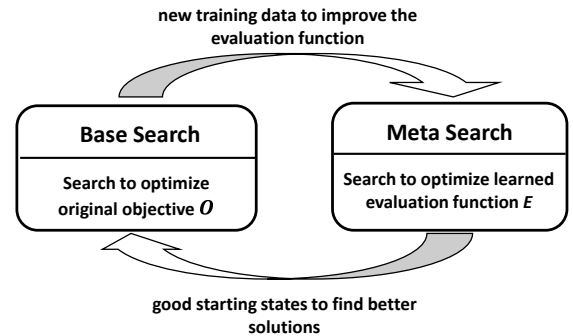


Fig. 1: High-level overview of the optimization algorithm.

combines the NoC performance metrics namely the network latency and energy consumption per message in a principled manner. Optimizing the communication cost ensures lower average hop count and improvement in the network performance in terms of both latency and energy consumption. However, the space of physically feasible SW based 3D NoC designs  $D$  is combinatorial in nature and our goal is to find the design  $d \in D$  that minimizes  $O$ . One could employ search algorithms such as hill-climbing and simulated annealing, which are very popular in the design community for this task. However, we leverage machine learning techniques that are shown to outperform these local search algorithms to intelligently explore the design space [21]. This optimization process is undertaken before the actual NoC implementation.

#### A. Optimization based on Machine Learning

We employ an online learning algorithm called STAGE [21], which was originally developed to improve the performance of local search algorithms (e.g., hill climbing) with random-restarts for combinatorial optimization problems. The key insight behind STAGE is to leverage some extra features  $\phi(d) \in R^m$  ( $m$  is the number of features) of the optimization problem to learn an improved evaluation function  $E$  that can estimate the promise of a design  $d$  as a starting point for the local search procedure  $A$ . It employs  $E$  to intelligently select promising starting states that will guide  $A$  towards significantly better solutions. Past work in the search community concluded that many practical optimization problems exhibit a “globally convex” or “big valley” structure, where the set of local optima appear convex with one global optimum in the center [22]. The main advantage of STAGE over popular algorithms such as simulated annealing is that it tries to learn the solution space structure, and uses this information cleverly to explore a much bigger design space in the given time. *To the best of our knowledge, this is the first work that applies STAGE to an NoC design optimization problem.*

The algorithm repeatedly alternates between two types of search as shown in Fig. 1: 1) Base search, where  $A$  is run with the original objective  $O$  until it reaches a local optima and new training data is generated to improve  $E$ ; and 2) Meta search, where it performs search with the learned evaluation function  $E$  to select good starting states to improve the performance of the local search procedure  $A$ . We want to learn  $E$  such that the estimated value of design  $d$  is equal to the expected best objective ( $O$ ) value seen on a search trajectory that starts from design  $d$  and follows the local search method  $A$  guided by  $O$ . In the initial exploration phase,  $E$  may not lead to good solutions but as the iterations progress,  $E$  will improve with the training data generated from the search experience in base search mode.

The effectiveness of the learned  $E$  depends on a small subset of critical training examples that successfully teach how to avoid different local optima during the meta-search phase. The STAGE algorithm tries to quickly identify this critical set in an adaptive manner.

We initialize  $E$ , training set  $Z$ , and initial design  $d_0$ . The following high-level algorithmic steps of STAGE are repeated for several iterations.

**Base search using  $A$  guided by  $O$ :** From  $d_0$ , run the search procedure  $A$  until a local optima is reached thereby leading to a search trajectory  $(d_0, d_1, \dots, d_T)$ .

**Improve  $E$ :** For each design  $d_i$  on the search path, add  $(\phi(d_i), y_i)$  to  $Z$ , where  $y_i$  is the best value along the search. Re-train  $E$  using a regression learner  $R$  with the updated training set  $Z$ .

**Meta search guided by  $E$ :** Continue from  $d_T$  and optimize  $E$  by performing a hill-climbing search to produce the best predicted starting state  $\hat{d}$ . If  $\hat{d}$  is the same as  $d_T$  (no search progress), set  $d_0$  to a random design. Otherwise, set  $d_0 = \hat{d}$ .

At the end, we return the best design found over all the iterations.

### B. Instantiation for 3D NoC Optimization

In this section, we provide all the details needed to apply the STAGE algorithm to our 3D NoC optimization problem.

**Design Space:** Our design space depends on a set of network resources, which are given as input to the optimization algorithm. These resources are defined as follows. 1) Cores ( $C$ ): A set of all cores  $C = \{C_1, C_2, \dots, C_N\}$ , where  $N$  is total number of cores. We assume that every core is connected to at least one router; 2) Planar Dies ( $P$ ): A set of all dies  $P$ . For  $N = 64$ , we consider four dies with each die containing 16 cores. For core placement, we follow a greedy algorithm to minimize  $(f_{ij} * d_{ij})$ , where  $f_{ij}$  and  $d_{ij}$  are the communication frequency and Cartesian distance between the cores respectively. In this step, we form clusters with 16 cores in each die; 3) Link Distribution ( $L$ ): The link length distribution  $L = \{l_1, l_2, \dots, l_k\}$ , where  $k$  depends on the size and topology of the network;  $l_i$ 's are determined based on the SW connectivity parameter  $\alpha$ . For higher values of  $\alpha$ ,  $l_k$  decreases; and 4) Communication Frequency ( $F$ ): The communication frequency among different cores  $F = \{f_{ij} \mid 1 \leq i, j \leq N, i \neq j\}$ . We assume that  $F$  for each application is given as an input to perform application-specific network optimization.

The set of all physically realizable SW NoC designs with the given link distribution  $L$  forms our design space.

**Objective Function  $O$ :** We define  $O$  as the communication cost of the given 3D NoC, which is the product of hop count, frequency of communication, and link length summed over every source and destination pair, i.e.,

$$O = \sum_{i=1}^N \sum_{j=1, i \neq j}^N (r * h_{ij} + d_{ij}) * f_{ij} \quad (1)$$

where  $f_{ij}$ , and  $d_{ij}$  are defined as above;  $h_{ij}$  is hop count between  $i$ - and  $j$ -th node, and  $r$  denotes the number of router stages. From a practical point-of-view,  $r$  is the number of cycles a message spends inside a router to move from input to output port. An NoC design with low  $O$  will have low latency and energy consumption, and hence, low energy-delay-product (EDP).

**Network Constraints:** To explore only physically feasible 3D NoC designs, we enforce some constraints on the placement of vertical links and router configurations. If TSVs are

considered as the vertical links, we only allow placing them point-to-point (regularly) between the routers. Such constraints may put additional limits on the performance of NoC designs. However, efficient optimization can overcome such limitations. The SW network has an irregular connectivity. Hence, the number of links connected to each router is not constant. For fair comparison between our SW network and 3D MESH, we assume that both of them use the same average number of connections,  $\langle k_{avg} \rangle$  per router. This also ensures that the 3D SW NoC does not introduce additional links compared to a 3D MESH. For a 64-core system,  $\langle k_{avg} \rangle$  is 4.5 considering all the routers, including the peripheral ones. In addition, the maximum connectivity per node,  $\langle k_{max} \rangle$ , is set to be 7 for the SW network as found in [23].

**Starting States and Successor Function:** For starting states, we randomly generate a SW network that satisfies the network constraints. The successor function  $S$  takes a network as input and returns a set of next states, and allows the search procedure to navigate the NoC design space.  $S$  generates one candidate state for each link connecting two nodes in the input network. It simply removes that link and places a link with the same length between two nodes in the network that are not directly connected.

The STAGE algorithm can benefit if we can specify the starting state distribution using some domain knowledge. Therefore, we also consider a starting-state distribution named  $\alpha$ -Greedy. We formulate the starting state (design) construction as a sequential decision-making task, where we select the next link to be placed at each step. In  $\alpha$ -Greedy distribution, we select a link greedily with probability  $\alpha$  based on communication frequency and a random link with probability  $(1-\alpha)$ . We start with  $\alpha=1$  (completely greedy) and gradually reduce  $\alpha$  to increase the randomness.

**Local Search Procedure  $A$ :** We employed a stochastic hill-climbing procedure, where the next states are sampled stochastically.

**Feature Function  $\phi$ :** The main challenge in adapting STAGE to our NoC domain is to define a set of features  $\phi$  for each network that can drive the learner. We divide the whole network into several overlapping subgraphs or regions, and define a set of features that can be categorized into three types: 1) Average hop count ( $h$ ), which calculates the average hop count for each region or sub-network; 2) Weighted communication which is defined as the sum of the products of hop count and communication frequency over all source-destination pairs for a particular hop count ( $\sum_{i=1}^N \sum_{j=1, j \neq i}^N f_{ij} * h_k$ ). The highest value of  $k$  depends on the network size and topology. If the value of this feature is small, it indicates that highly communicating cores are placed in the same neighborhood; and 3) Clustering coefficient ( $C_c$ ), which captures the connectivity of one core with its neighbors [24]. While the hop count takes into account mainly long-range communication, the clustering coefficient focuses more on local connectivity among the immediate neighbors. We found these features to sufficiently capture the network characteristics, efficient to compute, and allow to learn highly accurate evaluation function,  $E$ .

**Regression Learner:** The quality of our optimization methodology depends on the accuracy of the evaluation function  $E$ . In this work, we employ the support vector regression (SVR)

learner to learn  $E$  from the training data generated while running the STAGE algorithm.

Our training data consists of a set of input-output pairs  $\{(x_i, y_i)\}_{i=1}^n$ , where each  $x_i \in R^m$  is a feature vector and  $y_i \in R$  is the corresponding output. The  $\varepsilon$ -SVR algorithm tries to learn a function  $E$  such that the deviation of the predicted output  $E(x_i)$  from correct output  $y_i$  is less than the error tolerance  $\varepsilon$ .

Without loss of generality, we assume  $E$  is a linear function of the form  $E = \langle w, x \rangle + b$ , and present the  $\varepsilon$ -SVR formulation in primal form:

$$\min: \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n (\xi_i + \xi_i^*) \quad (2)$$

$$\text{S. t.} \quad \begin{cases} y_i - \langle w, x_i \rangle - b \leq \varepsilon + \xi_i \\ \langle w, x_i \rangle + b - y_i \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases} \quad (3)$$

In supervised learning, the goal is to learn a function that will perform well on unseen examples (generalization) and not the one that minimizes the error on the training data (over-fitting). This problem is generally addressed by adding a penalty term to discourage complex functions. In Equation (2), the first part is the penalty term (regularizer) and the second part is the training error.  $C$  is the regularization parameter that provides the tradeoff between minimizing the training error and generalization to unseen data, and  $\xi_i$  and  $\xi_i^*$  are slack variables to handle infeasible constraints. Linear functions won't suffice for complex problems such as ours. Therefore, we employ the radial basis function (RBF) kernel ( $K(x, x') = e^{-\gamma \|x-x'\|^2}$ ) to be able to learn non-linear functions, where  $\gamma$  is a tuning parameter. We selected the RBF kernel over other kernels because of its flexibility and predictive power. Additionally, to find the best learned function, we need to search over different values of  $(\varepsilon, C, \gamma)$ . We employ LibSVM [25] to learn the regression function and select the best combination of  $(\varepsilon, C, \gamma)$  over training set  $Z$  via the inbuilt  $v$ -fold ( $v=5$  in this work) cross-validation approach.

#### IV. EXPERIMENTAL RESULTS AND ANALYSIS

In this section, we present the performance of our optimized 3D SW NoC architecture. For this performance evaluation, we consider three metrics: latency, energy consumption, and energy-delay-product (EDP). The EDP is defined as the product of network latency and energy consumption, and unifies both of them into a single parameter. We also present a comparative performance evaluation of the 3D SW NoC with respect to other existing regular and irregular counterparts.

##### A. Experimental Setup

To evaluate the performance of different NoCs, we use a cycle-accurate NoC simulator that can simulate any regular or irregular 3D architecture. Our system consists of 64 cores or 64 network routers equally partitioned in four layers. The length of each packet is 64 flits and each flit consists of 32 bits. The routers are synthesized from an RTL level design using TSMC 65-nm CMOS process in Synopsys<sup>TM</sup> Design Vision. All router ports have a buffer depth of two flits and each router port has four virtual channels in case of irregular NoC. The NoC simulator uses wormhole routing, where the data flits follow the header flits once a path is established by the router. For regular 3D mesh-based NoC, XYZ-dimension order based routing is used. For irregular architectures such as the SW network, the

TABLE I: FEATURE DESCRIPTION

Feature Type	Feature Count
Avg. hop count for nine overlapping regions	9
Weighted communication ( $\sum \sum f_{ij} * h_k$ ) considering maximum hop count for N=64	8
Avg. Clustering coefficient (C <sub>c</sub> ) for four planar dies	4

topology-agnostic Adaptive Layered Shortest Path Routing (ALASH) algorithm is adopted [26]. The energy consumption of the network routers, inclusive of the routing strategies, was obtained from the synthesized netlist by running Synopsys<sup>TM</sup> Prime Power, while the energy dissipated by wireline links was obtained through HSPICE simulations, taking into consideration the length of the wireline links. We consider four SPLASH-2 benchmarks, namely, FFT, RADIX, LU, and WATER [27], and five PARSEC benchmarks, namely, DEDUP, VIPS, FLUIDANIMATE, CANNEAL, and BODYTRACK (BT) [28] in this performance evaluation.

In this work, for  $N = 64$  cores, we divide the whole network into nine regions. For each region, we consider average hop counts as the features. In addition, the initial network has the highest hop count of eight, and hence we require eight features for weighted communication cost. Finally, for each die in the network, we consider the average clustering coefficient and it gives rise to four more features. Table I lists all these features.

##### B. Performance of the Optimization Algorithm

In Section 3, we described the details of the optimization algorithm and how it is applied to design the 3D SW NoC architecture. Here we first characterize the performance of the optimization algorithm by quantifying various performance metrics of the 3D SW NoC.

###### 1) Communication Cost ( $O$ ) and Prediction Error

We create the initial network following the power law distribution shown in Section 3, where long-range links are placed randomly. Our goal is to find an optimized network starting from this random SW network. We call this initial NoC architecture as **3D SW<sub>random</sub>**. Fig. 2 shows the communication cost of the optimized network as a function of the number of iterations of the STAGE algorithm and the corresponding prediction error (in %) of the learned function. During this process, the learned function  $E$  predicts an initial network configuration to start the local search procedure that can lead to lower communication cost ( $O$ ). As seen from the figure, during the initial exploration phase, the error-rate is non-monotonic and high. After a few iterations the prediction error reduces to less than 1%, and after 20 iterations, the error is almost zero (0.05%). The prediction error remained more or less the same for all the subsequent iterations. These results indicate the effectiveness of our network features  $\phi$  and the SVR learning algorithm. Note that the best  $O$ -value decreases monotonically as the set of explored designs increases over the iterations. We also ran the same experiment with the  $\alpha$ -Greedy starting state distribution as mentioned above. However, the communication cost  $O$  and the prediction error have similar characteristics as the random distribution for the benchmarks and the system size considered in this work. Therefore, we present and discuss our results with random starting-state distribution.

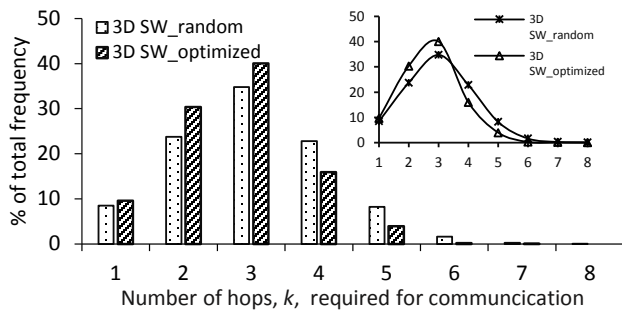


Fig. 3. Effect of optimization algorithm on weighted communication features.

The learned evaluation function  $E$  becomes highly accurate after a small number of iterations, and produces good starting states to help the local search procedure  $A$  in producing optimized network architectures with lower objective  $O$  (communication cost). We denote the final optimized NoC as **3D SW\_optimized**.

### 2) Characteristics of the Design: Random vs. Optimized

Now we investigate why the STAGE based optimization algorithm is suitable for developing energy-efficient NoC architectures. In Section 3.3, we described the details of the feature definition ( $\phi$ ), to represent each network. So, we will see how the design features change before and after optimization. Here we specifically consider the role of the weighted communication feature mentioned in Section 3.1.1. Fig. 3 shows the weighted communication feature, which reveals the percentage of total communication that is constrained between two nodes separated by  $k$  hops ( $k \geq 1$ ). Careful observation of Fig. 3 shows that for 3D SW\_optimized, the traffic constrained within one, two, and three hop communication increases compared to 3D SW\_random. Moreover, the amount of traffic that has to traverse beyond three hops decreases.

Hence, the inter-node communication that takes place in less than three hops becomes more frequent. Since the average hop count of the optimized network is calculated to be 2.94, any communication below this average hop count can be considered as efficient. Essentially, optimized network becomes more efficient for the same objective function.

The inset in Fig. 3 shows the percentage of communication versus the number of hops, where the area under the curve

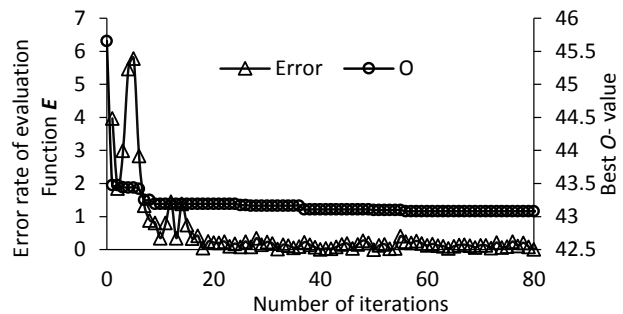


Fig. 2. Best objective value  $O$  and error rate of the evaluation function  $E$  for the STAGE algorithm over iterations

denotes the weighted communication feature mentioned in section 3.2. We can see that the 3D SW\_optimized curve shifts towards the left, which means that on an average any message in the optimized network traverses less hops compared to the initial network. Hence, it spends less time inside the network and occupies less network resources. Therefore, the STAGE-based optimization algorithm guides the search to converge to an efficient architecture.

### C. Effect of Optimization on 3D SW NoC

In this section, we evaluate and compare the performances of the 3D SW\_optimized and the un-optimized 3D SW\_random architectures. For comparison purpose, all the values are normalized with respect to 3D MESH.

#### 1) Network Latency

Fig. 4.a demonstrates the effect of optimization for 3D SW NoC. The optimization improves the network latency on an average of 3% over the un-optimized version, and 5.5% over the conventional 3D MESH. The optimization process redistributes the links among the cores such that cores that have to frequently communicate with each other are either directly connected or need to traverse a small number of hops. This results in reduced average hop count and weighted communication for 3D SW\_optimized NoC. A new set of benchmarks with higher injection rates will highlight the benefits of this work even more, which is the focus of our future work.

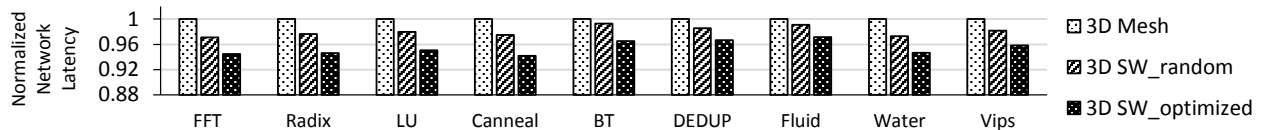


Fig. 4.a: Normalized network latency of 3D SW before and after optimization

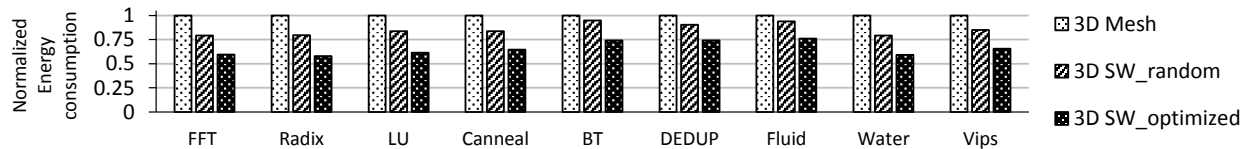


Fig. 4.b: Normalized energy consumption per message of 3D SW before and after optimization

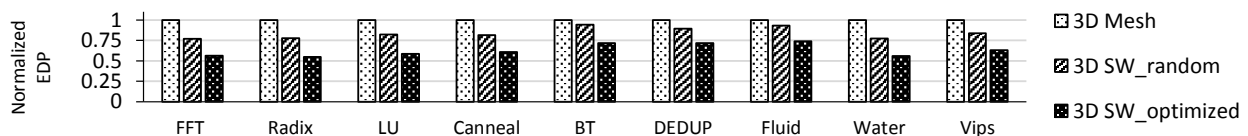


Fig. 4.c: Normalized energy-delay-product (EDP) of 3D SW before and after optimization

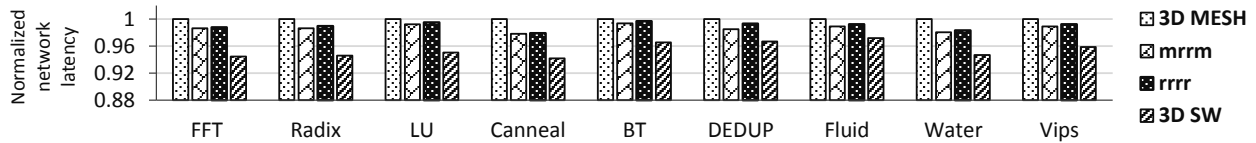


Fig. 5.a: Normalized network latency of different 3D NoCs

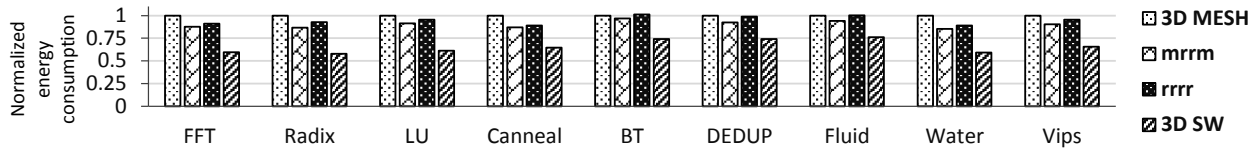


Fig. 5.b: Normalized energy consumption per message of different 3D NoCs

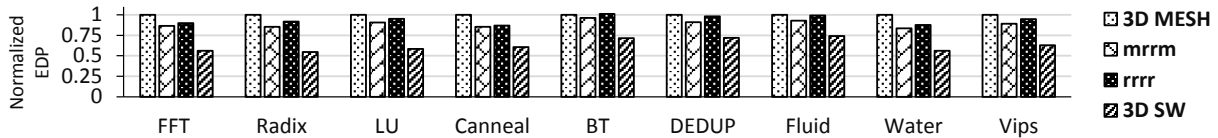


Fig. 5.c: Normalized EDP for different 3D NoCs

## 2) Energy Consumption

Energy consumption per message depends on the energy consumed by the router as well as the planar and vertical links.

The STAGE-based optimization algorithm reduces average hop count and communication cost, which contributes to the minimization of the router and link energy consumption respectively. Fig. 4.b plots the energy consumption profile before and after optimization normalized to these values for the 3D MESH. On an average, an optimized 3D SW NoC shows 33% and 17% energy consumption improvement over the 3D MESH and 3D SW\_random respectively. Fig. 3 helps us in understanding the reasons behind the improvement in energy consumption. The area under the 3D SW\_optimized curve is less than that of the un-optimized counterpart. Hence 3D SW\_optimized reduces the utilization of network resources for any message. As a result, both the router and link energy decrease and the overall energy consumption profile improves.

## 3) Energy-delay-Product (EDP)

From the EDP profile shown in Fig. 4.c, we observe that the average EDP of 3D SW NoC is reduced by approximately 35% and 19% compared to 3D MESH and 3D SW\_random respectively. The improvement in the energy-delay product is direct consequence of the improvement in network latency and energy consumption of the 3D SW\_optimized NoC.

## D. Comparative Performance Evaluation

In the previous sections, we showed that 3D SW\_optimized significantly outperformed 3D SW\_random. In this section, we compare 3D SW\_optimized with several existing 3D NoC architectures. Henceforth, we refer 3D SW\_optimized as 3D SW for simplicity. For this comparative performance evaluation, we consider 3D MESH and two recently proposed irregular 3D NoCs, namely, *mrrm* and *rrrr* [29]. Both the *mrrm* and *rrrr* NoCs have point-to-point vertical connections as in 3D MESH and 3D SW. However, their die-level planar connection pattern varies. For *rrrr*, all the four dies have randomly connected interconnection patterns. On the other hand, *mrrm* has random connection patterns in the middle two dies whereas the first and the fourth dies follow mesh-based regular connectivity. To build *mrrm* and *rrrr*, we follow the method suggested in [29] and keep the number of links equal to that of 3D MESH and 3D

TABLE II. COMPARISON OF AVERAGE HOP COUNT AND COMMUNICATION COST OF 3D NOC ARCHITECTURES

NoC architecture	Avg. hop count	Communication cost, $\mathcal{O}$
3D SW	2.94	43.08
<i>mrrm</i>	3.07	47.27
<i>rrrr</i>	3.03	47.39
3D MESH	3.81	55.5

SW. All the performance metric values are normalized with respect to the 3D MESH.

### 1) Network Latency

Fig. 5.a shows the network latency of all the 3D NoCs. Among all the NoCs, 3D MESH and 3D SW exhibit the highest and the lowest latency respectively. The other two architectures namely *mrrm* and *rrrr* perform somewhere in the middle. As in the case of 3D SW NoC, both *mrrm* and *rrrr* have irregularities in the horizontal planes. However, the number and the length of the links are not optimized for these architectures. For *rrrr*, the link distribution has large number of long-range links that help communication among long-distant cores at the expense of nearby communication. In the case of 3D SW NoC, the link distribution follows the power law and the connection pattern is optimized to facilitate both the nearby and long-range communications.

The *mrrm* architecture maintains the link distribution in between *rrrr* and 3D SW NoC. Hence, its network latency lies in between *rrrr* and 3D SW. Finally, 3D MESH NoC suffers from higher average hop count compared to other 3D architectures due to multi-hop communication pattern; hence, it suffers from the highest network latency. Table 2 lists the communication costs and average hop counts for all these NoCs. As expected, 3D SW and 3D MESH exhibit the lowest and highest communication cost and hop count respectively, whereas *mrrm* and *rrrr* reside in between these two. The effect of these communication costs is eventually reflected in the latency characteristics.

### 2) Energy Consumption

Fig. 5.b shows the energy consumption per message for different 3D NoCs. Among all these, 3D MESH has highest energy consumption followed by *mrrm*, *rrrr* and 3D SW NoC.

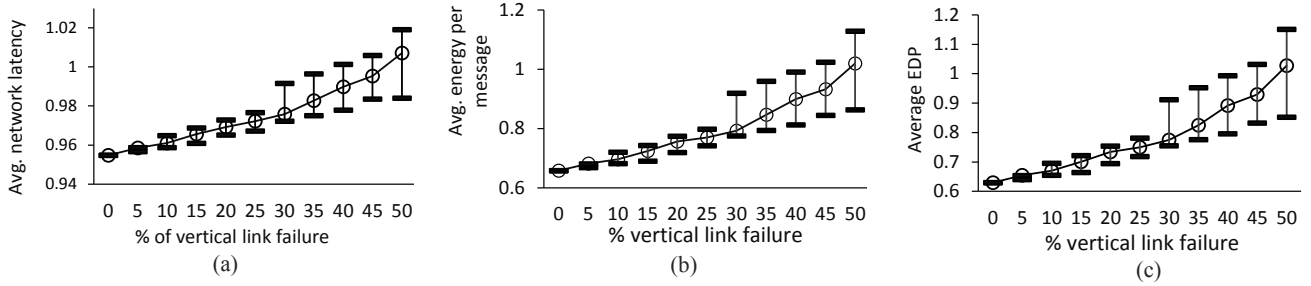


Fig. 6. 3D SW NoC performances normalized to fault free (fully vertical connected) 3D MESH Vs. the percentage of vertical link failure rate. (a) Average normalized network latency; (b) Average energy consumption per message; and (c) Average EDP

Higher network latency gives rise to higher network resources utilization and hence, higher energy consumption per message. For 3D MESH, the router energy consumption is significantly higher due to multi hop communication, so it performs the worst among all of them. The *mrrm* and *rrrr* NoCs are capable of reducing the router energy consumption compared to mesh and performs better than 3D MESH. However, due to their random link distribution, they suffer from higher communication cost and average hop count compared to the optimized SW NoC. Hence, they consume more link energy and router energy. With the least communication cost, 3D SW NoC consumes the lowest energy possible among all these architectures.

### 3) Energy-delay-Product (EDP)

The energy-delay-product is directly affected by network latency and energy consumption. The architecture that performs best in terms of latency and energy consumption is expected to have lower EDP compared to the others. Fig. 5.c presents the EDP profile of different 3D NoCs. As expected, 3D SW NoC has the lowest EDP profile followed by *mrrm*, *rrrr* and 3D MESH. On an average, 3D SW has 35% lower EDP profile compared to 3D MESH while the highest improvement of 43% was found for RADIX.

## V. ROBUSTNESS OF 3D SW NOC ARCHITECTURE

In this section, we analyze the robustness of the 3D SW NoC architecture under vertical link failure. The reason behind studying the scenario of vertical link failures is that despite the recent advancements in the TSV technology, TSVs are still subject to failure due to voids, cracks, and misalignment [30].

In this case, after building and optimizing the 3D SW NoC, the vertical links are randomly removed to simulate the link failure scenario. This principally tests the robustness of the SW interconnection network. Starting from the fault free 3D SW NoC, we increase the link failure percentage with a step size of 5% till 50% of the vertical links are randomly removed. In Fig. 6, we show the average network latency, energy consumption, and energy-delay-product (EDP) for all the benchmarks by varying the amount of failed links. All parameters are again normalized to the values of fault-free fully connected 3D MESH. The figures show the worst, best and average

performance levels for the same amount of link failures over 1000 different runs.

From these figures, we observe that as the link failure percentage increases, the network latency, energy consumption, and EDP increase gradually. The difference between the worst- and best case scenarios also increases progressively. This occurs due to the fact that with increasing vertical link failure, we have fewer routing resources than what is required to achieve optimum performance. In addition, for the case of 50% vertical link failure, the average network latency and energy consumption almost equal the corresponding values for a fully connected fault-free 3D MESH. For this case, it is also quite striking that EDP matches with that of the fault-free 3D MESH as well. In the worst case, the network latency shows 2% higher value whereas, the energy consumption and EDP record 13% and 15% higher values respectively compared to fault free 3D MESH. These results therefore show that even with significantly fewer number of vertical links, the 3D SW NoC performs as well as the fully connected fault free 3D MESH on average. It has been observed in [19][20] that SW networks display remarkable resilience to high rates of link failures since the average distance between nodes in a SW network increases by a small margin with the rate of failures. Hence, the effect of vertical link failures on the performance is minimal.

### A. Optimization Quality with less Resources

We also analyze the quality of the proposed optimization algorithm in the presence of limited resources. To do so, we consider the 3D SW NoC with 50% vertical links and optimize the placement of the links following our STAGE algorithm-based methodology. We show that the optimized NoC with reduced vertical links can still maintain a high-level of performance if the limited resources (vertical links in this case) are utilized optimally. For the rest of the work, we denote this optimized architecture as **3D SW<sub>partial</sub>** and compare its performance with respect to the fully connected 3D MESH.

Fig. 7 shows the latency, energy, and EDP for 3D SW<sub>partial</sub> with respect to the fully connected 3D MESH. We observe that on an average, 3D SW<sub>partial</sub> still shows 2.5% lower latency compared to the fully connected 3D MESH. In comparison to the results for the fully connected 3D SW NoC shown in Fig. 5.a, 3D SW<sub>partial</sub> incurs only 3% higher network

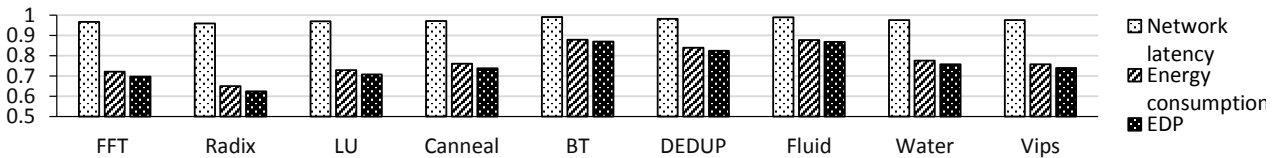


Fig. 7. Network latency, energy consumption and EDP per message of the 3D SW<sub>partial</sub>. All the values are normalized to that of fault free (fully connected) 3D MESH

latency. The reason for this behavior is that the optimization algorithm ensures the most suitable link placement considering the available resources. As a result, the latency penalty remains low. Similarly, 3D SW<sub>partial</sub> shows 24% and 25% lower energy and EDP respectively compared to the fully connected 3D MESH. In addition, by comparing these energy and EDP values with Fig. 5.b and 5.c for the fully connected 3D SW, we see that 3D SW<sub>partial</sub> pays only 12% and 13% penalty respectively over its fully connected counterpart.

This result carries the promise of highly energy-efficient design with reduced resources. For 3D SW<sub>partial</sub> NoC, we have reduced 50% of the vertical links, which are predominantly long-range shortcuts and still, on an average, we incur no more than 15% penalty. The algorithm optimizes the link distribution among the cores such that the overall communication cost is minimized. Hence, the 3D SW NoC with reduced number of vertical links utilizes its resources very efficiently to compensate for the resource reduction.

If we compare the performances of 3D SW NoC with 50% vertical link failure without any optimization (Section 4.5) with the optimized 3D SW<sub>partial</sub>, then we find that the later performs better in every performance metric compared to the former. On an average, for latency, energy, and EDP metrics, 3D SW<sub>partial</sub> shows improvements of 2.5%, 27%, and 26% respectively compared to its un-optimized counterpart. Hence, the optimization algorithm plays a crucial role in minimizing the performance penalty due to the limited resources. We can emphatically conclude that the proposed 3D SW NoC along with the optimization methodology is robust enough to compensate for the performance loss due to vertical link reduction. Overall, the performance penalty is small compared to the proportion of resource reduction.

## VI. CONCLUSIONS

We proposed a robust design optimization methodology to improve the energy efficiency of 3D NoC architectures by combining the benefits of SW networks and machine learning techniques to intelligently explore the design space. We showed that the optimized 3D SW NoC architecture outperforms existing 3D NoCs. The optimized 3D SW NoC on an average achieves 35% EDP reduction over conventional 3D MESH. We also demonstrated the efficacy and robustness of our optimization methodology by producing 3D SW NoC architectures that can perform equally or better than the fully connected 3D MESH with significantly less number of vertical links. For the case of 50% reduction in vertical links, the optimized 3D SW NoC achieves 25% lower EDP compared to fully vertically connected 3D MESH NoC.

## ACKNOWLEDGEMENT

This work was supported in part by the US National Science Foundation (NSF) grants CCF-0845504, CNS-1059289, and CCF-1162202, and Army Research Office grant W911NF-12-1-0373.

## REFERENCES

- [1] V. Pavlidis and E. Friedman, *Three-Dimensional Integrated Circuit Design*, V. Pavlidis and E. Friedman, Eds. Morgan Kaufmann, 2009.
- [2] B. S. Feero, P.P. Pande "Networks-on-Chip in a Three-Dimensional Environment: A Performance Evaluation", IEEE Trans. on Computers, Vol 53, pp. 32-45, Aug. 2008.
- [3] P. Jacob et al., "Predicting the Performance of a 3D Processor-Memory Stack," IEEE Design and Test of Computers, vol. 22, no. 6, pp. 540-547, Nov./Dec. 2005.

- [4] H.G. Lee et al., "On-Chip Communication Architecture Exploration: A Quantitative Evaluation of Point-to-Point, Bus, and Network-on-Chip Approaches," ACM Trans. Design Auto. of Electronic Sys., vol. 12, no. 3, pp. 1-20, Aug. 2007.
- [5] F. Li et al., "Design and Management of 3D Chip Multi-processors Using Network-in-Memory," Proc. of ISCA, 2006 pp. 130-141, doi>10.1109/ISCA.2006.18.
- [6] J. Kim et al., "A novel dimensionally-decomposed router for on-chip communication in 3D architectures," in Proc. of Annual Intl. Symp.on Comp. Arch., June 2007, pp. 138-149.
- [7] A. M. Rahmani, K. R. Vaddina, K. Latif, L. P. Plosila, H. Tenhunen, "High-Performance and Fault-Tolerant 3D NoC-Bus Hybrid Architecture Using ARB-NET Based Adaptive Monitoring Platform", IEEE Transaction on Computers, vol 63, pp. 734-747, Mar 2014.
- [8] L. Jiang, F. Ye, Q. Xu, K. Chakrabarty, B. Eklow, "On effective and efficient in-field TSV repair for stacked 3D ICs", Proc. of DAC. 2013. DOI=10.1145/2463209.2488824
- [9] M. Bahmani, A. Sheibanyrad, F. Petrot, F. Dubois, P. Durante, "A 3D-NoC Router Implementation Exploiting Vertically-Partially-Connected Topologies," Annual Symp. on VLSI, Aug. 2012, vol., no., pp.9-14.
- [10] I. Loi, F. Angiolini, S. Mitra, L. Benini, "Characterization and Implementation of Fault-Tolerant Vertical Links for 3-D Networks-on-Chip", IEEE Trans. Computer Aided Design of Integrated Circuit and Sys., vol 30, pp. 124-134, Jan. 2011.
- [11] S. Murali, C. Seiculescu, L. Benini and G. De Micheli, "Synthesis of Networks on Chips for 3D Systems on Chips", Proc. of ASPDAC, Jan. 2009, pp. 242-247.
- [12] K. Wang, S. Dong, "Power optimization for application-specific 3D network-on-chip with multiple supply voltages" Proc. of ASPDAC., Jan. 2009, pp. 362 - 367.
- [13] K. Srinivasan, K. S. Chatha, "A methodology for layout aware design and optimization of custom network-on-chip architectures," ISQED, Mar. 2006, vol., no.6 pp.357, 27-29.
- [14] Y. Xu, Y. Du, B. Zhao, X. Zhou, Y. Zhang, J. Yang, "A low-radix and low-diameter 3D interconnection network design," Symp. On HPCA, Feb. 2009, pp.30-42.
- [15] A.W. Topol, et al. "Three-Dimensional Integrated Circuits," IBM Jour. of Research and Dev. Vol. 50, nos. 4, July 2006.
- [16] C. Macron et al., "Tiny NoC: A 3D Mesh Topology with Router Channel Optimization for Area and Latency Minimization", Intl Conf on VLSI Design, 2014, pp 228-233.
- [17] S. Manipatruni et al. "Wide temperature range operation of micrometer-scale silicon electro-optic modulators," Opt. Lett. Vol. 33, Issue 19, pp. 2185-2187, (2008).
- [18] U. Y. Ogras, R. Marculescu, "'It's a Small World After All': NoC Performance Optimization via Long-Range Link Insertion", IEEE Trans. on VLSI Systems, Vol. 14, pp 693-706, 2006.
- [19] C. Teuscher, "Nature-inspired interconnects for self-assembled large-scale network-on-chip designs", e-print arXiv: 0704.2852.
- [20] T. Petermann, P. D. L. Rios, "Spatial small-world networks: a wiring-cost perspective", e-print arXiv:cond-mat/0501420.
- [21] J. A. Boyan, Andrew W. Moore: Learning Evaluation Functions to Improve Optimization by Local Search. Journal of Machine Learning Research 1: 77-112 (2000).
- [22] K. D. Boese, Cost versus distance in the traveling salesman problem. Technical Report CSD-950018, UCLA Computer Science Dept, 1995.
- [23] R. Kim et al., "Energy-Efficient VFI-Partitioned Multicore Design Using Wireless NoC Architectures", In Proc. of Intl Conf on, CASES 2014, Article No 3, New York, USA.
- [24] M. D. Humphries, K. Gurney, "Network 'Small-World-Ness': A Quantitative Method for Determining Canonical Network Equivalence" Journal of PLOS One, April 30, 2008, DOI: 10.1371/journal.pone.0002051
- [25] LibSVM Software: <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>
- [26] O. Lysne, T. Skeie, S. A. Reinemo, I. Theiss, "Layered Routing in Irregular Networks", IEEE Transaction on Parallel and Distributed System, vol. 17, pp. 51- 65, 2006.
- [27] S. C. Woo, M. Ohara, E. Torrie, J. P. Singh, A. Gupta, "The SPLASH-2 programs: characterization and methodological considerations," In Proc. of International Symposium on Computer Architecture, 1995, pp. 24-36.
- [28] C. Bienia, "Benchmarking modern multiprocessors," Ph.D. Dissertation, Princeton Univ., Princeton NJ, Jan. 2011.
- [29] H. Matsutani et al., "Low-Latency Wireless 3D NoCs via Randomized Shortcut Chips", Proc. of DATE, 2014, pp. 1-6.
- [30] B. Noia and K. Chakrabarty, Design-for-Test and Test Optimization Techniques for TSV-based 3D Stacked ICs, Springer, 2014.