Plane-Balanced and Deadlock-Free Adaptive Routing for 3D Networks-on-Chip

Nizar Dahir*, Ra’ed Al-Dujaily, Alex Yakovlev and Petros Missailidis
School of Electrical and Electronic Engineering
Newcastle University
Newcastle upon Tyne, UK
{nizar.dahir, raaed.aldujaili, alex.yakovlev, petros.missailidis}@ncl.ac.uk

Terrence Mak
Department of Computer Science and Engineering
The Chinese University of Hong Kong
Shatin, Hong Kong
stmak@cse.cuhk.edu.hk

ABSTRACT
This paper proposes a new method for designing adaptive routing algorithms for 3D networks-on-chip (NoCs). This method is based on extending the existing 2D turn model adaptive routing to a 3D scenario. A 3-D plane-balanced approach with maximal degree of adaptiveness is achieved by applying a well-defined set of rules for different strata of the 3D NoC. The proposed method is applicable to any of the turn models. In this paper, we employ odd-even turn model as a basis for introducing the proposed strategy. Experimental results show that the new 3D odd-even can achieve up to 23.8% improvement in throughput over conventional odd-even approach. The improvement is consistent for different traffic types. The proposed method enables a new avenue to explore adaptive approaches for future large-scale 3D integration.

Categories and Subject Descriptors
C.2.1 [Network Architecture and Design]: Network communications

Keywords
Networks-on-Chip, 3D IC, Routing Algorithms, Degree of Adaptiveness, Dynamic Programming

1. INTRODUCTION
Networks-on-chip (NoC) as a communication paradigm of Systems-on-Chip (SoC) and Chip Multiprocessors (CMPs) tackles many challenges associated with on-chip communication by providing better scalability, higher flexibility and power efficiency in integrating many intellectual property (IP) cores in a single chip [1].

On the other hand, semiconductor manufacturing processes are approaching the physical limits. This brought the attention to explore using three-dimensional (3D) VLSI design which brings tremendous advantages. Shorter global interconnects lengths, less delay, better scalability; heterogeneous integration and smaller form factor are among these advantages. 3D integration can be achieved by stacking 2D dies vertically and vertical communication is achieved using through-silicon-vias (TSVs) [2, 7]. Using NoCs with 3D multi core systems enable the integration of unprecedented number of cores for future 3D CMPs.

Extending NoC to the third dimension opened a new area of research. One of the main challenges for 3D NoCs is the design of efficient routing algorithm. Routing algorithms efficiency can be measured using different metrics. For example, routing algorithms for 3D NoCs can be designed with the aim to achieve higher fault tolerance as in [10,12]. Also, 3D routing may target minimizing packet delays and, thus, maximizing performance as proposed in [11].

One of the important metrics used for NoC routing algorithm evaluation is the degree of adaptiveness or path diversity. The degree of adaptiveness is defined as the number of possible paths between a source and destination. It can significantly impact performance of adaptive routing algorithms. For instance, assuming minimal routing, using dimension-order routing (e.g. XY routing for 2D meshes or XYZ for 3D meshes) results in deterministic routing with degree of adaptiveness equal to one. Conversely, Fully-Adaptive routing has the maximum degree of adaptiveness. However, Fully-Adaptive routing cannot guarantee deadlock freeness; thus, deadlock detection and recovery techniques are required in this case. These techniques have their power and area overhead and can impact performance. To avoid this overhead, the turn model is used where some turns are prohibited to guarantee the deadlock freeness. Examples are West-First, North-Last and Negative-First [6] for 2D meshes.

Turn model routing algorithms result in uneven degree of adaptiveness due to the dependence of the path diversity on the packet direction. Thus, the adaptiveness is not distributed uniformly across the network. To achieve more balanced degree of adaptiveness odd-even routing has been proposed in [5]. Different rules are used for the odd and
even columns in a 2D mesh to achieve more even adaptiveness than the turn model. The authors in [3], also, suggested extending the odd-even routing to 3D meshes by applying the same rules for all 2D layers in addition to a rule to prohibit deadlocks that involve channels of the 3rd dimension.

Turn model or odd-even routing algorithms can be readily extended to 3D by applying their rules in 2D and a rule for the 3rd dimension as suggested in [3]. However, extension in this way will result in the same rules applied for all layers. In this work we propose a new method of extending 2D partially adaptive routing algorithms to 3D. The method is based on applying different (complimentary) rules for different layers. This is shown to give more even degree of adaptiveness which is reflected as higher performance in terms of throughput and average delay when a runtime global optimization is considered. The runtime optimization technique used in this work to evaluate the proposed method is Dynamic Programming Networks (DPN).

Dynamic Programming Network (DPN) has been recently proposed as a mean of adaptive optimal path routing in Networks-on-Chip (NoC) [8, 9]. DPN guides packets to the minimum cost path among the available paths between source and destination. The degree of adaptiveness (or path diversity) offered by the routing algorithms has a crucial impact on DPN performance. In this work we will use DPN as selection strategy for runtime global optimization to evaluate our proposed 3D routing algorithm.

The main contribution of this paper is introducing a new approach for extending 2D mesh partially adaptive routing algorithms to 3D. This method results in plane-balanced degree of adaptiveness by applying different rules for different layers. Although the principle can be applied to any of the turn models, for convenience, it will be described in the context of odd-even routing. Evaluation of the proposed method under different traffic scenarios is performed by comparison with conventional 3D routing. Experiments show that the proposed method achieves higher performance under various traffic scenarios.

2. BACKGROUND

In this section definitions of concepts and terms used in this paper are given. Moreover, this section reviews the original odd-even for 2D meshes and its extension to 3D meshes. Also Dynamic Programming Networks will be reviewed.

2.1 Definitions

In the section definition of terms and concepts that are going to be used throughout this paper are given. The term turn is used and is defined as follows:

**Definition 1.** A turn in a 3D mesh consists of any 90-degree change in the packet direction and consists of combination of any two of the three channels: row, column and pillar such that the tail node of one of the channels is the head node of the other.

Also, the terms odd and even are used to identify row, column or plane. This is illustrated as follows:

**Definition 2.** In a 3D mesh, a column is called an even (respectively, odd) column if the dimension-0 coordinate of the column is an even (respectively, odd) number. A row is called an even (respectively, odd) row if the dimension-1 coordinate of the row is an even (respectively, odd). Similarly, a plane is called an even (respectively, odd) plane if the dimension-2 coordinate of the plane is an even (respectively, odd).

2.2 The Odd-Even Turn Model

The odd-even routing is shown to give higher degree of adaptiveness compared to other turn model routing algorithms [3]. The odd-even routing prohibits turns to break the waiting cycles and prevent deadlocks. However, the odd-even differ from the turn model in that it prohibits different turns for odd and even columns. This is shown to give higher degree of adaptiveness compared to other turn model routing algorithms [3]. In summary the rules of odd-even in 2D meshes are [3]:

- **Rule 1:** In odd column packets are not allowed to take North-West (NW) turns nor South-West (SW) turns,
- **Rule 2:** In even column packets are not allowed to take East-North (EN) nor East-South (ES) turns.

The possible eight turns and two cycles that can occur in 2D mesh are shown in Fig. 1. Rules 1 and 2 ensures that the column of NW turn cannot have EN turn and the column of the SW turns cannot have ES turns. This ensures the deadlock freeness by breaking all possible cycles as depicted by the dashed lines in Fig. 1.

![Illustration of the prohibited turns for the odd-even routing (rules 1 and 2). Dashed lines represent prohibited turns.](image)

2.3 Dynamic Programming Networks

Minimum cost path (or shortest path) problem is crucial to NoCs run-time management. A range of run-time issues, such as dynamic routing, fault tolerance design and thermal management, can be formulated as shortest path problems and resolved using dynamic programming network [8]. The shortest path problem can be described as follows: Given a directed graph $G = (V, A)$ with $n = |V|$ nodes, $m = |A|$ edges, and a cost $C_{u,v}$ associated with each edge $u, v \in A$. The total cost of a path $p = (n_0, n_1, \ldots, n_k)$ is the sum of the costs of its constituent edges: $Cost(p) = \sum_{i=1}^{k} C_{n_{i-1}, n_i}$. The shortest path from $n_{n_0}$ to $n_{n_k}$ is then defined as the optimal path ($p^*$) which can be expressed as:

$$p^* = \arg \min_{p \in \{P_{n_{n_0}}, n_{n_k}\}} \{Cost(p)\} \quad (1)$$

where $\{P_{n_{n_0}}, n_{n_k}\}$ is the set of all possible paths between $n_{n_0}$ and $n_{n_k}$. The shortest path problem can be stated in the form of Bellman equations, which defines a recursive procedure in step $k$ and can lead to a simple parallel architecture.
to speed up the computation. Finding the shortest path from \(n_0\) to \(n_w\), requires the notion of DP value or namely cost-to-go, which is the expected cost from \(n_0\) to \(n_w\). This cost is updated recursively until it reaches its optimal value.

This algorithm is known as dynamic programming. We denote the DP value for \(n_0\) to \(n_w\) at the \(k\)-th iteration as \(V^{(k)}(v, w)\). \(V^*(v, w)\) is the optimal DP value which equals to the resolved variable \(C(v, w)\). The Bellman equation becomes:

\[
V^{(k)}(v, w) = \min_{u \in V} \left\{ V^{(k-1)}(u, w) + C_{v,u} \right\}
\]

where \(V(w, w) = 0\). If the recursion is expanded from \(n_0\) to \(n_k\), the optimal DP-value can be expressed as the minimum cost of the path from node \(n_0\) to node \(n_k\).

\[
V_k(n_0, n_k) = \min_{\{n_0, n_1, \ldots, n_k\} \in P_{n_0, n_k}} \left\{ \sum_{i=1}^{k} C_{i-1,i} \right\}
\]

where destination node \(n_w = n_k\) and \(P_{n_0, n_k}\) is the set of paths from \(n_0\) to \(n_k\), all of which have \(k\) edges. Moreover, the optimal decision at any node \(n_i\) along the shortest path can be readily obtained from the argument of the minimum operator as follows:

\[
\mu(u, w) = \text{arg min}_{v \in V} \{ V^*(u, w) + C_{v,u} \}
\]

where the optimal decision becomes \(\mu(v, w)\). The dynamic programming cost computation can be implemented using distributed dynamic programming units (DPUs). Assuming a multi-source single destination, each unit gets the costs of the neighboring units as input and propagate the minimum cost after adding its local cost. In this work we aim to route the packets to the least congested route minimize the packet delay and maximize performance. To achieve this, the number of routed flits in the router is used as a cost for that node. Thus, the cost will be associated with nodes rather than edges.

For partially-adaptive routing cost propagation and computation must consider only the possible directions i.e. directions allowed by the routing algorithm. In this scenario the degree of adaptiveness offered by the routing algorithm plays a crucial role in the DPN performance.

### 3. ROUTING IN A 3-D NETWORK-ON-CHIP

#### 3.1 3D Odd-Even Routing

Extending the 2D odd-even model (described by rules 1 and 2) to 3D meshes requires applying a rule to ensure deadlock freeness and prohibit waiting cycles which consists of vertical turns (turns involving Up or Down directions). To achieve this, the following rule is used:

- **Rule 3**: \(xy\) – Down turns are not allowed in an odd \(xy\)-plane and \(Up\) – \(xy\) turns are not allowed in an even \(xy\)-plane.

In other words, packets travelling upward cannot enter and even \(xy\)-plane (turn North, East, South or West) and packets travelling within an odd \(xy\)-plane cannot leave this plane through the downward direction. This ensures that no plane can have both \(Up\) – to – \(xy\) and \(xy\) – to – \(Down\) and, thus, break any possible waiting cycles that involve vertical channels and ensure deadlock freeness. This rule is illustrated in Fig. 2. In the following the 3D odd-even routing that results from applying rules 1 and 2 and rule 3 will be called Conventional OE. A proof of deadlock freeness Conventional OE can be readily derived from the proof of deadlock freeness of 2D odd-even routing for 2D meshes presented in [3].

#### 3.2 Plane-Balanced 3D Odd-Even Routing

Before describing the proposed 3D odd-even, we will define a modified odd-even model. In contrast to the Conventional OE, the rules of the modified model involve turn prohibitions that are applied according to the row and not column. The rules of the modified odd-even can be stated as:

- **Rule 4**: In odd row packets are not allowed to take West-North (WN) turns nor East-North (EN) turns.
- **Rule 5**: In even row packets are not allowed to take South-West (SW) turns nor South-East (SE) turns.

![Illustration of the prohibited vertical turns for the 3D odd-even routing (rule 3).](image)

![Illustration of the prohibited turns in the modified odd-even routing (rules 4 and 5).](image)
Corollary 1. In 3D NoCs deadlock freeness is still guaranteed when different layers have different turn prohibition rules if these rules guarantee intra-layer deadlock freeness and a rule is applied to guarantee freeness of deadlocks that involve vertical turns.

Corollary 1 can be readily deduced from the proof of deadlock freeness of turn model [4] or odd-even model [3].

For conventional 3D odd-even routing, odd-even rules within xy-plane are the same for all xy-planes. They are applied along the column. Based on Corollary 1 different rules can be used for different layers to achieve more balanced adaptiveness. The conventional 3D odd-even is modified such that the rules for odd xy-plane (z coordinate is odd) are different from even xy-plane (z coordinate is even). The 3D routing algorithm proposed in this work applies rules 1 and 2 in odd layer and rules 4 and 5 in even layer. This is in addition to rule 3. The resulting 3D odd-even algorithm will be called Balanced Odd Even.

![Figure 4: Illustration of path diversities for both, (a) conventional 3D odd-even, and (b) the proposed balanced 3D odd-even.](image)

3.3 Degree of Adaptness

The degree of adaptiveness is one of the metrics that are used to evaluate partially adaptive routing algorithms [4]. It can be defined as the number of allowable different paths from a source to a destination. For 3D mesh let the coordinates of the source node be \((x_s, y_s, z_s)\) and the coordinates of destination node be \((x_d, y_d, z_d)\). Moreover, let \(\Delta x = x_d - x_s\), \(\Delta y = y_d - y_s\) and \(\Delta z = z_d - z_s\). Also, in the following, let \(d_x = |\Delta x|\), \(d_y = |\Delta y|\) and \(d_z = |\Delta z|\).

For fully adaptive routing, the degree of adaptiveness is the number of all shortest paths from source to destination and is given by:

\[
P_{\text{full adaptive}} = \frac{(d_x + d_y + d_z)!}{d_x!d_y!d_z!} \tag{5}\]

The degree of adaptiveness of Conventional OE can be expressed as follows:

\[
P_{\text{Conventional OE}} = \frac{(h + d_y + k)!}{h!d_y!k!} \tag{6}\]

where \(h\) is equal to \(\lceil \frac{\Delta x}{d_x} \rceil\) or \(\lceil \frac{\Delta x}{d_y} \rceil\) depending on the column at which \(x_s\) lies and \(d_x\). Similarly \(k\) is equal to \(\lceil \frac{\Delta x}{d_z} \rceil\) or \(\lceil \frac{\Delta x}{d_y} \rceil\) depending on the layer at which \(z_s\) lies and \(d_z\). It can be noticed that for Conventional OE the constrained directions are \(x\) and \(z\) while the \(y\) direction is relaxed. On the other hand, the degree of adaptiveness of Modified OE can be expressed as:

\[
P_{\text{Modified OE}} = \frac{(d_x + q + k)!}{d_x!q!k!} \tag{7}\]

where \(q\) is equal to \(\lceil \frac{\Delta y}{d_y} \rceil\) or \(\lceil \frac{\Delta y}{d_z} \rceil\) depending on the row at which \(y_s\) lies and \(d_y\). In contrary to Conventional OE (Eq. (6)), it can be noticed that for Modified OE the constrained directions are \(y\) and \(z\) while the \(x\) direction is relaxed.

Conventional OE and Modified OE have different relaxation of directions (Equations (6) and (7), respectively). For the proposed Balanced odd-even routing, applying Conventional OE for for odd layers and Modified OE for even layers will result in the restrictions in the odd xy-plane (x direction is constrained as indicated by Eq. (7)). Consequently, the regularity of the traffic patterns (and the resulting communication workload) that occurs for adjacent layers (due to similar restrictions) is broken as shown in Fig 4. This will lead to a more balanced adaptiveness among the planes which enhances the performance of any runtime adaptive selection strategy (such as DPN).

4. RESULTS AND DISCUSSION

To evaluate the proposed routing technique for 3D NoCs, a 3D NoC-based CMP is considered. The floorplan is arranged as a 3D mesh with size of \(6 \times 6 \times 4\). The traffics used in our experiments are; Uniform, Transpose, and Hotspot. For the Random traffic each tile sends data to all other tiles with equal probability. For the Transpose case tile \((i, j, k)\) sends packets to tile \((N_x - i, N_y - j, N_z - k)\) where \(N_x\), \(N_y\) and \(N_z\) are the NoC dimensions in \(x\), \(y\) and \(z\) directions, respectively. In the Hotspot traffic pattern the four central tiles of the top layer (layer 0) receive an extra 5% in addition to the Uniform (random) traffic.

Traffic simulation is performed using a modified version of Noxim [5]. The router architecture is modified to support 3D NoCs. Moreover, the 2D NoC routing algorithms and traffics are modified to support the 3D NoC routings and traffics. Packet size is assumed to be 8 flits and buffer size is 4 flits. As we mentioned earlier, a DPN guided selection strategy will be used to evaluate the efficacy of the proposed 3D routing algorithms. Thus, at a particular node, the routing logic will return a set of directions that the packet can take and the DPU will chose the one with lowest cost to the destination. In this context the DPN will provide global optimization in which the degree of adaptiveness of the routing algorithm has direct impact on the performance of the DPN. The cost of the node for DP selection is considered to be the number of routed flits through this node. This would route packets through the least congested paths to minimize average delay and maximize performance. The following routing algorithms are compared:

- **Odd-Even(buffer):** Conventional OE rules are applied (Rule 1,2 and Rule 3 are applied for all planes) with buffer level selection strategy.
- **Odd-Even(DP):** Conventional OE with dynamic programming guided selection strategy to guide packets to the least congested path among the available paths between a source and a destination.
• **Balanced Odd-Even (DP):** The proposed Balanced OE routing in which, in addition to rule 3, rules 1 and 2 are applied in an odd $xy$-plane and rules 4 and 5 are applied in an even $xy$-plane. Dynamic programming guided selection strategy is also used in this case.

Fig. 5 compares the performance of the three algorithms for the Random traffic case in terms of average network delay versus packet injection rate (PIR), Fig. 5a, and throughput versus PIR, Fig. 5b. It can be noticed that, using DPN resulted in significant improvement over buffer level selection strategy for the same routing algorithm (Conventional OE). However, our balanced odd-even with DPN outperforms both Conventional OE with buffer level and Conventional OE with DPN under this traffic scenario.

![Figure 5](image)

**Figure 5:** Comparison of the performances of the three algorithms for Random traffic.

Figures 6 and 7 plot performance metrics (throughput and delay) under the Transpose and Hotspot traffics, respectively. It is evident from these results that balanced odd-even improvement over Conventional OE is consistent for all traffic types. However, we can see that this improvement is higher for some traffics than others.

Tables 1 reports a summary of the results in terms of Saturation Point Throughput achieved by the network for the three routing algorithms and three traffic scenarios. The Saturation Point Throughput were for each traffic scenario and for each algorithm is computed as the throughput at highest PIR at which the network is not yet saturated. It also shows the percent improvement of proposed Balanced OE (DP) over Odd-Even (Buffer), and Odd-Even (DP). It can be noticed the proposed algorithm outperforms the conventional odd-even with buffer selection by up to 23.8 percent. In the case when the same selection strategy is used (i.e. DP), it can be seen that the improvement is up to 8.3 percent.

The consistent improvement of balanced odd-even suggest the effectiveness of the proposed method of extending 2D partially adaptive algorithms. It achieves a more balanced degree of adaptiveness compared to conventional extension method which translates as higher throughput and lower average delay for different traffic scenarios.

![Figure 6](image)

**Figure 6:** Comparison of the performances of the three algorithms for Transpose traffic.

5. **CONCLUSION**

In this work a novel method for extending turn model adaptive routing algorithms from 2D to 3D NoCs is proposed. The method applies different rules for different layers which results in different restriction on traffic flow for different layers. Using our approach, a 3-D plane-balanced approach with higher degree of adaptiveness is achieved. Although the proposed method is applicable to any of the turn models, for convenience, we described it in the context of odd-even routing. Path diversity analysis and deadlock freeness of the proposed method are discussed and compared to the conventional 3D odd-even method. Experimental results show that the proposed balanced odd-even with DPN can achieve improvement of up to 23.8% compared odd-even with buffer level and 8.3% compared to odd-even with
Table 1: Improvement in the Saturation Point Throughput of Balanced OE(DP) for Different Traffic Scenarios.

<table>
<thead>
<tr>
<th>Traffic</th>
<th>Saturation point throughput (flit/cycle/IP)</th>
<th>Balance OE(DP) improvement vs. Odd-Even (Buffer)</th>
<th>vs. Odd-Even (DP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Odd-Even</td>
<td>Odd-Even (Buffer)</td>
<td>Odd-Even (DP)</td>
<td>Balanced OE(DP)</td>
</tr>
<tr>
<td>Random</td>
<td>0.11</td>
<td>0.125</td>
<td>0.1352</td>
</tr>
<tr>
<td>Transpose</td>
<td>0.105</td>
<td>0.122</td>
<td>0.13</td>
</tr>
<tr>
<td>Hotspot</td>
<td>0.105</td>
<td>0.108</td>
<td>0.117</td>
</tr>
</tbody>
</table>

Figure 7: Comparison of the performances of the three algorithms for Hotspot traffic.

DPN. Moreover, the improvement is consistent for all the considered traffic types. This suggests the effectiveness of the proposed method in improving chip’s adaptivity which is directly reflected as higher performance for runtime adaptive routing that uses dynamic programming networks. This work reveals a new opportunity for future 3D networks-on-chip routing adaptivity.

6. REFERENCES