Abstract—The increasing viability of three-dimensional (3D) silicon integration technology has opened new opportunities for chip architecture innovations. One direction is in the extension of two-dimensional (2D) mesh-based tiled chip-multiprocessor architectures into three dimensions. In this paper, we focus on efficient routing algorithms for such 3D mesh networks. As in the case of 2D mesh networks, throughput and latency are important design metrics for routing algorithms. Existing routing algorithms suffer from either poor worst-case throughput (DOR [1], ROMM [3]) or poor latency (VAL [2]). Although the minimal routing algorithm O1TURN proposed in [4] already achieves near-optimal worst-case throughput for the 2D case, the optimality result does not extend to higher dimensions. For 3D and higher dimensional meshes, the worst-case throughput of O1TURN degrades tremendously. The main contribution of this paper is the design of a new oblivious routing algorithm called Randomized Partially-Minimal (RPM) routing. RPM provably achieves optimal worst-case throughput for 3D meshes when the network radix \( k \) is even and within a factor of \( 1/k^2 \) of optimal worst-case throughput when \( k \) is odd. RPM also outperforms VAL, DOR, ROMM, and O1TURN in average-case throughput by 33.3%, 111%, 47%, and 30%, respectively when averaged over one million random traffic patterns on an \( 8 \times 8 \times 8 \) topology. Finally, whereas VAL achieves optimal worst-case throughput at a penalty factor of 2 in average latency over DOR, RPM achieves (near) optimal worst-case throughput with a much smaller factor of 1.33. In practice, the average latency of RPM is expected to be closer to minimal routing because 3D mesh networks are not expected to be symmetric in 3D chip designs. The number of available device layers is expected to be much less than the number of processor tiles that can be placed along an edge of a device layer. For practical asymmetric 3D mesh configurations, the average latency of RPM reduces to just a factor of 1.11 of DOR.

I. INTRODUCTION

There has been considerable discussion in recent years on the benefits of three-dimensional (3D) silicon integration in which multiple device layers are stacked on top of each other with direct vertical interconnects tunneling through them [13], [14], [15], [16], [17]. 3D integration promises to address many of the key challenges that arise from the semiconductor industry’s relentless push into the deep nanoscale regime. Recent advances in 3D technology in the area of heat dissipation and micro-cooling mechanisms have alleviated thermal concerns regarding stacked device layers. Among the benefits, 3D integration promises the ability to provide huge amounts of communication bandwidth between device layers and integrate disparate technologies in the same chip design.

The increasing viability of 3D technology has opened new opportunities for chip architecture innovations. One direction is in the extension of two-dimensional (2D) tiled chip-multiprocessor architectures [9], [10], [11], [12] into three dimensions [18], [19]. Many proposed 2D tiled chip-multiprocessor architectures have relied on a 2D mesh network topology as the underlying communication fabric. Extending mesh-based tiled chip-multiprocessor architectures into three dimensions represents a natural progression for exploiting 3D integration. The focus of this paper is on providing efficient routing for such 3D mesh networks.

As in the case of 2D mesh networks, throughput and latency are important performance metrics in the design of routing algorithms. Ideally, a routing algorithm should maximize both the worst-case and average-case throughput and minimize the length of routing paths. Although dimension-ordered routing (DOR) algorithm [1] achieves minimal-length routing, it suffers from poor worst-case and average-case throughput because it offers no route diversity. On the other hand, the routing algorithm proposed by Valiant (VAL) [2] achieves optimal worst-case throughput by load balancing globally across the entire network. However, it suffers from poor average-case throughput and long routing paths. ROMM [3] provides another alternative that achieves minimal routing and good average-case throughput by considering route diversity in the minimal direction, but it suffers from poor worst-case throughput.

For the case of 2D mesh networks, Seo et al. [4] described a novel routing algorithm called O1TURN that achieves both minimal-length routing and near-optimal worst-case throughput. O1TURN simply chooses between two possible minimal-turn paths (XY and YX) for routing. Despite the simplicity, it was shown that O1TURN achieves optimal worst-case throughput when the network radix \( k \) is even and within a factor of \( 1/k^2 \) of optimal worst-case throughput when \( k \) is odd. However, as observed in [4], the near-optimal worst-case throughput property of O1TURN does not extend to higher dimensions.\(^1\) Perhaps surprisingly, the worst-case throughput of O1TURN degrades tremendously for higher dimensional meshes. For example, in the 3D case for an \( 8 \times 8 \times 8 \) mesh, the worst-case throughput of O1TURN degrades to just 30% of optimal. The corresponding worst-case throughput values for DOR and ROMM are even less at around 13% and 26% of optimal, respectively.

In this paper, we introduce a new oblivious routing algorithm called Randomized Partially-Minimal (RPM) rout-

\(^1\) Although technically the 3D version of O1TURN is called “O2TURN”, we will simply refer to the algorithm as O1TURN so that the same name can be applied to all higher dimensional meshes.
ing that achieves near-optimal worst-case throughput, good average-case throughput, and good average latency. Conceptually, RPM works as follows: In the 3D case, suppose we use Z to denote the “vertical” dimension and XY to denote the two “horizontal” dimensions. RPM works by first routing a packet in the minimal direction to a random intermediate “plane” in the vertical dimension; i.e., it first routes a packet in the minimal direction to a random intermediate Z position. It then routes the packet on the XY plane using either minimal XY or YX routing. Finally, it routes the packet in the minimal direction in the Z vertical dimension to its final destination. The entire Z-XY-Z or Z-YX-Z path makes at most three turns. Effectively, RPM load-balances traffic uniformly across all k vertical layers and routes traffic minimally only in the two horizontal dimensions.

The main contributions of RPM are as follows:

(a) RPM provably achieves optimal worst-case throughput for 3D meshes when the network radix \( k \) is even and within a factor of \( 1/k^2 \) of optimal worst-case throughput when \( k \) is odd.

(b) RPM outperforms VAL, DOR, ROMM, and O1TURN in average-case throughput by 33.3\%, 111\%, 47\%, and 30\%, respectively, when averaged over one million random traffic patterns for an \( 8 \times 8 \times 8 \) mesh network.

(c) On average latency, as measured in network hops, RPM does not achieve minimal-length routing because non-minimal routing is used in one of three dimensions. However, whereas VAL achieves optimal worst-case throughput at a penalty factor in average latency of 2 over DOR, RPM achieves (near) optimal worst-case throughput with a much smaller factor of 1.33.

(d) In practice, the average latency of RPM is expected to be closer to minimal routing because 3D mesh networks are not expected to be symmetric in 3D chip designs. In particular, the number of available device layers is expected to be much less than the number of processor tiles that can be placed along an edge of a device layer. For example, for a four layered \( 16 \times 16 \times 4 \) mesh, the average latency of RPM reduces to just a factor of 1.11 of DOR.

The rest of the paper is organized as follows: Section II provides a brief background on performance metrics. Section III describes the RPM routing algorithm and presents analytical results. Section IV describes extensions to RPM for higher dimensional and asymmetric meshes. Section V evaluates RPM’s performance. Section VI concludes the paper.

II. BACKGROUND

In this section, we provide a brief overview of analytical methods used to evaluate worst-case and average-case throughput. In particular, we elaborate on the concept of network capacity and a method to compute worst-case throughput for oblivious algorithms. We then elaborate on a method to compute average-case throughput. To simplify the discussion on throughput analysis, we ignore flow control issues, and we assume single flit packets that route from node to node in a single cycle.

Network capacity is defined by the maximum channel load \( \gamma^* \) that a channel at the bisection of the network needs to sustain under uniformly distributed traffic. For any \( k \)-ary \( n \)-mesh [21], independent of \( n \),

\[
\gamma^* = \begin{cases} \frac{k}{4^\frac{k-1}{2}} & \text{if } k \text{ is even} \\ \frac{k}{4^\frac{k-1}{2} - 1} & \text{if } k \text{ is odd} \end{cases} \tag{1}
\]

The network capacity is the inverse of \( \gamma^* \).

The maximum channel load \( \gamma(R, \Lambda) \) for a routing algorithm \( R \) and traffic matrix \( \Lambda \) is the expected traffic load crossing the heaviest loaded channel under \( R \) and \( \Lambda \). The worst-case channel load \( \gamma_{wc}(R) \) for a routing algorithm \( R \) is the heaviest channel load that can be caused by any admissible traffic. Admissible traffic is defined to be any doubly sub-stochastic matrix \( \Lambda \) with all row and column sums bounded by 1. Suppose a network consists of \( N \) nodes, a traffic matrix \( \Lambda = (\lambda_{ij}) \) is an \( N \times N \) matrix where \( \lambda_{ij} \) represents the expected traffic from node \( i \) to node \( j \). The traffic matrix \( \Lambda \) is doubly sub-stochastic and hence admissible if

\[
\sum_{i=1}^{N} \lambda_{ij} \leq 1, \forall j \text{ and } \sum_{j=1}^{N} \lambda_{ij} \leq 1, \forall i
\]

and it is said to be doubly stochastic if

\[
\sum_{i=1}^{N} \lambda_{ij} = 1, \forall j \text{ and } \sum_{j=1}^{N} \lambda_{ij} = 1, \forall i
\]

As shown in [5], an admissible traffic matrix that can cause the worst-case channel load for a routing algorithm \( R \) can be found by solving a derived maximum weighted matching problem. The worst-case saturation throughput for a routing algorithm \( R \) is the inverse of the worst-case channel load. The normalized worst-case saturation throughput, \( \Theta_{wc}(R) \), is defined as the worst-case saturation throughput normalized to the network capacity:

\[
\Theta_{wc}(R) = \left( \frac{\gamma_{wc}(R)}{\gamma^*} \right)^{-1} \tag{2}
\]

Unless otherwise noted, we will simply refer to \( \Theta_{wc}(R) \) as the worst-case throughput of \( R \).

Using the methodology used in [5], [4], the average-case throughput for a routing algorithm \( R \) can be computed by averaging the throughput over \( T \), a large set of random traffic patterns:

\[
\Theta_{avg}(R) = \frac{1}{|T|} \sum_{\Lambda \in T} \left( \frac{\gamma(R, \Lambda)}{\gamma^*} \right)^{-1} \tag{3}
\]

In this paper, we used \( |T| = \text{one million} \).
III. RANDOMIZED PARTIALLY-MINIMAL ROUTING

The basic idea behind RPM is rather simple. Conceptually, RPM works by load-balancing flits uniformly across all $k$ vertical layers along the $Z$ dimension, just like VAL [2], but only along one dimension. RPM then routes flits on each XY plane using minimal XY or YX routing with equal probability. Finally, RPM routes flits to their final destinations along the $Z$ dimension. Figure 1 depicts two possible RPM routing paths. In particular, let $(x_1,y_1,z_1)$ be the source, $(x_2,y_2,z_2)$ be the destination, and $\hat{z}$ be the randomly chosen intermediate $Z$ position. The two corresponding Z-XY-Z and Z-YX-Z routing paths are $(x_1,y_1,z_1) \rightarrow (x_1,y_1,\hat{z}) \rightarrow (x_2,y_2,\hat{z}) \rightarrow (x_2,y_2,z_2)$ and $(x_1,y_1,z_1) \rightarrow (x_1,y_1,\hat{z}) \rightarrow (x_1,y_2,\hat{z}) \rightarrow (x_2,y_2,z_2)$, respectively, with at most three turns. When $x_1 = x_2$ and $y_1 = y_2$, then the traffic is just uniformly randomized along the $Z$ dimension. In this case, when $\hat{z}$ is greater than both $z_1$ and $z_2$, or when $\hat{z}$ is less than both $z_1$ and $z_2$, a loop is formed in the path $z_1 \rightarrow \hat{z} \rightarrow z_2$. These loops can be removed online before routing a packet to achieve a reduction in hop count. When the source and destination are the same, no routing is necessary. It should be noted that although we used load-balancing along the $Z$ dimension for this description, RPM can be equivalently defined by load-balancing uniformly along any one dimension and routing minimally in the remaining two dimensions.

A. Throughput Analysis

In this section, we prove that RPM achieves optimal worst-case throughput when the network radix is even ($k$ in a $k \times k \times k$ mesh network) and within a factor of $1/k^2$ of optimal when $k$ is odd. Since the $1/k^2$ term diminishes quadratically, the worst-case throughput of RPM when $k$ is odd rapidly converges to optimal with increasing $k$. We prove this near-optimality in three parts. We first prove that for any doubly sub-stochastic traffic matrix $\Lambda$, RPM’s uniform load-balancing on the $Z$ vertical dimension will guarantee that each traffic matrix $\hat{\Lambda}$ for a corresponding horizontal $k \times k$ plane is also doubly sub-stochastic. We next prove that the worst-case channel load on each XY plane is bounded by $k/2$, meaning that the worst-case channel load on any channel in the $X$ or $Y$ dimension is bounded by $k/2$. We also prove that the worst-case channel load for the vertical channels is bounded by $k/2$. Finally, using the expression shown in Equation 2 for computing the worst-case throughput, we characterize the near optimal nature of RPM.

Claim 1: Given any 3D doubly sub-stochastic traffic matrix, the 2D traffic that will traverse any XY plane using RPM will be doubly sub-stochastic.

Proof: It suffices to show that for any doubly stochastic traffic matrix $\Lambda$, each corresponding 2D traffic matrix $\hat{\Lambda}$ will be doubly stochastic. Let $\hat{\Lambda}[(x_s,y_s,z_s),(x_d,y_d,z_d)]$ be the traffic from $(x_s,y_s,z_s)$ to $(x_d,y_d,z_d)$. By definition of doubly stochastic, the traffic from any source $(x_s,y_s,z_s)$ to any destination $(x_d,y_d,z_d)$ in $\Lambda$ must sum to 1.

$$\sum_{z=0}^{k-1} \sum_{x=0}^{k-1} \sum_{y=0}^{k-1} \hat{\Lambda}[(x,y,z),(x,y,z)] = 1$$

Finally, using the two-phase load balancing of RPM in the $Z$ dimension, we have

$$\sum_{z=0}^{k-1} \sum_{x=0}^{k-1} \sum_{y=0}^{k-1} \hat{\Lambda}[(x,y,z),(x,y,z)] = 1$$

Let $\hat{\Lambda}$ be the 2D traffic matrix for a plane at some $z = \hat{z}$, and let $\hat{\Lambda}[(x_s,y_s),(x_d,y_d)]$ be the traffic between any two nodes on this plane. Given the two-phase load balancing of RPM in the $Z$ dimension, we have

$$\hat{\Lambda}[(x_s,y_s),(x_d,y_d)] = \sum_{z=0}^{k-1} \sum_{x=0}^{k-1} \sum_{y=0}^{k-1} \hat{\Lambda}[(x_s,y_s,z),(x_d,y_d,z)]$$

For $\hat{\Lambda}$ to be doubly stochastic, the row sum from any $(x_s,y_s)$ or the column sum to any $(x_d,y_d)$ in $\Lambda$ must all be 1.

$$\sum_{x=0}^{k-1} \hat{\Lambda}[(x_s,y_s),(x,y)] = 1$$

$$\sum_{y=0}^{k-1} \hat{\Lambda}[(x_s,y_s),(x,y)] = 1$$

Substituting Equations 6 and 4 into Equation 7, we have

$$\sum_{x=0}^{k-1} \sum_{y=0}^{k-1} \hat{\Lambda}[(x_s,y_s,z),(x,y,z)] = \frac{k-1}{k} = 1$$

Similarly, substituting Equations 6 and 5 into Equation 8, we have

$$\sum_{x=0}^{k-1} \sum_{y=0}^{k-1} \hat{\Lambda}[(x,y,z_s),(x_d,y_d,z)] = \frac{k-1}{k} = 1$$

Since all rows and columns in $\hat{\Lambda}$ sum to 1, the 2D traffic matrix $\hat{\Lambda}$ is doubly stochastic. This analysis holds for all $z \in [0,k).$
Claim 2: The maximum channel load in a $k \times k \times k$ mesh network with RPM routing is $k/2$.

Proof: By Claim 1, the 2D traffic on each XY plane is doubly sub-stochastic. It has already been shown in the context of 2D meshes that minimal XY and YX routing with equal probability results in a maximum channel load of $k/2$ [4]. Hence, it follows that the worst-case channel load on any channel in the X or Y dimension using RPM is bounded by $k/2$. Given the two-phase load balancing in the Z dimension, the one-dimensional (1D) traffic along any Z line is twice uniform. For meshes, the maximum channel load for uniform traffic is $k/4$ when $k$ is even and $(k^2 - 1)/4k$ when $k$ is odd. The maximum channel load for twice uniform is simply $2(k/4) = k/2$ when $k$ is even and $2((k^2 - 1)/4k) = (k^2 - 1)/2k$ when $k$ is odd. Since $k/2 \geq (k^2 - 1)/2k$, the worst-case channel load is bounded by $\gamma_{wc}(RPM) = k/2$. ■

Claim 3: RPM achieves optimal worst-case throughput when $k$ is even and within a factor of $1/k^2$ of optimal when $k$ is odd.

Proof: By Claim 2, $\gamma_{wc}(RPM) = k/2$. As reminded in Section II, $\gamma = k/4$ when $k$ is even. Therefore, using Equation 2, we have

$$\Theta_{wc}(RPM) = \left(\frac{k/2}{k/4}\right)^{-1} = 0.5$$

This is optimal since the optimal worst-case throughput has already been shown previously to be half of the network capacity [21]. When $k$ is odd, $\gamma = (k^2 - 1)/4k$. Therefore, we have

$$\Theta_{wc}(RPM) = \left(\frac{k/2}{(k^2 - 1)/4k}\right)^{-1} = \left(\frac{k}{2} \cdot \frac{4k}{(k^2 - 1)}\right)^{-1} = \frac{k^2 - 1}{2k^2} = 0.5 \left(1 - \frac{1}{k^2}\right)$$

This is within a factor of $1/k^2$ of optimal.

Figure 2 shows the worst-case throughput of RPM in comparison to VAL, DOR, ROMM, and O1TURN on 3D meshes with different network radices. Note that the performance of DOR, ROMM, and O1TURN all degrade tremendously with increasing radix. At $k = 14$, the worst-case throughput of RPM is 14 times higher than DOR and 5.26 times higher than both ROMM and O1TURN.

B. Latency Analysis

Let $H_{avg}(R)$ be the average latency for routing algorithm $R$ as measured in hop count. The average hop count calculation assumes that the traffic between all source-destination pairs are equal. For a $k \times k \times k$ mesh, the average hop count for DOR is

$$H_{avg}(DOR) = 3 \left(\frac{k^2 - 1}{3k}\right) = \frac{k^2 - 1}{k}$$

Each $(k^2 - 1)/3k$ component corresponds to the average hop count in each dimension using DOR [21].

In RPM, minimal routing is used in the X and Y dimensions resulting in an average hop count of $(k^2 - 1)/3k$ along each of these dimensions. When two-phase routing is used in the Z dimension without any loop removal, the average hop count for this dimension is twice that of DOR, namely $2[(k^2 - 1)/3k]$. However, as mentioned in Section III, a possibility of loop removal exists in the Z dimension when the X and Y coordinates of the source and destination are the same. Once loops are removed, two-phases of Z routing reduce to a single phase of minimal routing. The X and Y coordinates of source and destination nodes are equal with a probability of $1/k^2$. This results in the following expression for the average hop count of RPM:

$$H_x(RPM) = H_y(RPM) = \left(\frac{k^2 - 1}{3k}\right)$$

$$H_z(RPM) = \left(\frac{k^2 - 1}{k^2}\right) \times \left(\frac{2(k^2 - 1)}{3k}\right) + \frac{1}{k^2} \times \left(\frac{k^2 - 1}{3k}\right)$$

$$H_{avg}(RPM) = H_x(RPM) + H_y(RPM) + H_z(RPM) = \left(\frac{4}{3} \cdot \frac{1}{3k^2}\right) \times H_{avg}(DOR)$$

The penalty factor in average latency for using a partially minimal routing algorithm like RPM instead of a minimal routing algorithm like DOR can be quantified by computing the ratio of their average latencies. In particular, the latency ratio of two routing algorithms $R1$ and $R2$ is defined as

$$LR(R1, R2) = \frac{H_{avg}(R1)}{H_{avg}(R2)}$$

The penalty factor for RPM is simply

$$LR(RPM, DOR) = \frac{4}{3} \cdot \frac{1}{3k^2}$$

$LR(RPM, DOR)$ converges to 1.33 for relatively large values of $k$. This is much smaller than the penalty factor of $LR(VAL, DOR) = 2$ that VAL has to sacrifice to achieve optimal worst-case throughput.
C. Virtual Channels and Deadlocks

For 3D meshes, if RPM is implemented by load-balancing along the Z dimension, two virtual channels per physical channel is sufficient to achieve deadlock-free routing: one virtual channel for Z-XY-Z routing, and the other for Z-YX-Z routing. As described in section III, RPM can be equivalently defined using Y-ZX-Y and X-YZ-X and X-ZY-X paths. A randomized version of RPM can be defined which load-balances along each dimension with equal probability while routing minimally in the remaining two dimensions. The randomized version of RPM can be made deadlock-free using three virtual channels.

D. Asymmetric Meshes

Claim 4: RPM achieves optimal worst-case throughput when \( k_{\text{max}} = \max(k_x, k_y, k_z) = k_z \) and \( k_{\text{max}} \) is not equal to \( k_x \) or \( k_y \). When \( k_{\text{max}} = k_x \) or \( k_{\text{max}} = k_y \), RPM is optimal when \( k_{\text{max}} \) is even and is within a factor of \( 1/k_{\text{max}}^2 \) of optimal when \( k_{\text{max}} \) is odd.

Proof: Claim 1 already shows that the 2D traffic on any XY plane using RPM will be doubly sub-stochastic. Suppose \( k_{\text{max}} = \max(k_x, k_y, k_z) \) is equal to either \( k_x \) or \( k_y \). Since routing on the XY plane using RPM is the same as O1TURN, it follows that RPM achieves optimal worst-case throughput when \( k_{\text{max}} \) is even, and within a factor of \( 1/k_{\text{max}}^2 \) of optimal otherwise. If \( k_{\text{max}} = k_z \), then the worst-case throughput of RPM only depends on the maximum channel load in the Z dimension, which is optimal.

The latency analysis for asymmetric meshes is also very similar to the symmetric case. For an asymmetric \( k_x \times k_y \times k_z \) mesh, the average hop count for DOR is

\[
H_{\text{avg}}(\text{DOR}) = \left( \frac{k_x^2 - 1}{3k_x} \right) + \left( \frac{k_y^2 - 1}{3k_y} \right) + \left( \frac{k_z^2 - 1}{3k_z} \right) \tag{11}
\]

The average hop count for RPM is

\[
H_{\text{avg}}(\text{RPM}) = \left( \frac{k_x^2 - 1}{3k_x} \right) + \left( \frac{k_y^2 - 1}{3k_y} \right) + \left( 2 - \frac{1}{k_xk_y} \right) \left( \frac{k_z^2 - 1}{3k_z} \right) \tag{12}
\]

The latency ratio of RPM and DOR for the asymmetric case can be calculated using Equation 9.

IV. EXTENDING RPM TO HIGHER DIMENSIONS

In this section, we extend RPM to consider higher dimensional and asymmetric meshes. To avoid repeating the same analysis, we directly consider both extensions together. Let \( a_0, a_1, \ldots, a_{n-2}, a_{n-1} \) be the dimensions for an \( n \)-dimensional mesh, and let \( k_0, k_1, \ldots, k_{n-2}, k_{n-1} \) be the corresponding radices. RPM can be readily extended by first uniformly load-balancing flits across the last \( (n-2) \) dimensions using dimension-ordered routing, namely along \( a_2, a_3, \ldots, a_{n-2}, a_{n-1} \). RPM then routes flits minimally along the two dimensions \( a_0 \) and \( a_1 \) using either the \( a_0a_1 \) or \( a_1a_0 \) order with equal probability. Finally, RPM routes flits to their destinations along \( a_2, a_3, \ldots, a_{n-2}, a_{n-1} \) in dimension order.

Claim 5: RPM achieves optimal worst-case throughput when \( k_{\text{max}} = \max(k_0, k_1, \ldots, k_{n-1}) \) is equal to any one of \( k_2, k_3, \ldots, k_{n-1} \) and \( k_{\text{max}} \) is not equal to \( k_0 \) or \( k_1 \). If \( k_{\text{max}} = k_0 \) or \( k_{\text{max}} = k_1 \), RPM is optimal when \( k_{\text{max}} \) is even and within a factor of \( 1/k_{\text{max}}^2 \) of optimal when \( k_{\text{max}} \) is odd.

Proof: The proof for the above claim is a direct extension of the proof for Claim 4 and is not presented here to avoid repetition.

V. PERFORMANCE EVALUATION

In this section, we compare the performance of RPM with the routing algorithms described in Table I. The first four are oblivious routing algorithms that are independent of the network state. The last one is a deadlock-free minimal adaptive routing algorithm. We use the randomized version of RPM discussed in Section III-C for evaluating the performance of RPM on symmetric mesh topologies. For asymmetric meshes we load-balance only along the short vertical dimension. Randomization improves the average throughput of RPM on symmetric meshes while retaining the same worst-case throughput since it distributes traffic equally along all three dimensions.

We first perform a simplified throughput analysis that assumes an ideal single-cycle router with infinite buffers. We then back these results with more realistic flit-level simulations.

A. Throughput Evaluation

The normalized saturation throughput results (normalized to the network capacity) for each oblivious routing algorithm on each traffic pattern of Table II are shown in Table III for three different 3D mesh configurations. The traffic pattern described as DOR-WC is a worst-case traffic pattern for DOR. As discussed briefly in Section II, we use the methodology proposed in [5] to determine worst-case throughput. Since this worst-case throughput analysis is only applicable to oblivious routing algorithms, we defer to Section V-B to consider comparisons with a minimal adaptive routing algorithm using detailed flit-level simulations.

In particular, Table III shows normalized saturation throughput results for two symmetric mesh topologies and
one asymmetric topology. In practice, 3D mesh networks are not expected to be symmetric in 3D chip designs. The number of available device layers is expected to be much less than the number of processor tiles that can be placed along the edge of a device layer. Hence, we chose to evaluate the performance of RPM on a 16 × 16 × 4 mesh. The results validate that RPM indeed achieves (near) optimal worst-case throughput for both the symmetric and asymmetric cases.

For an 8 × 8 × 8 configuration, RPM outperforms VAL, DOR, ROMM, and O1TURN in average-case throughput by 33.3%, 111%, 47%, and 30%, respectively, which are significant improvements. The average case results were measured using the technique described in Section II using 1 million randomly generated traffic patterns. RPM also performs well under adversarial traffic patterns, namely on transpose and bit-complement traffic. Although DOR can achieve better normalized throughput than RPM when the traffic has already been uniformly randomized (which may not be true in practice), the results for DOR are significantly worse in the average and worst cases. On worst-case throughput, RPM achieves the same optimal worst-case throughput as VAL, but outperforms DOR, ROMM, and O1TURN by 694%, 279%, and 233%, respectively, on an 8 × 8 × 8 mesh.

For the asymmetric 16 × 16 × 4 mesh, RPM again achieves optimal worst-case throughput (same as VAL) and outperforms DOR, ROMM and O1TURN by 500%, 238% and 100% respectively. For this topology, RPM performs strictly better than all other routing algorithms considered, both in terms of average case throughput and throughput for the different traffic patterns except for uniform traffic where it is as good as DOR and O1TURN.

As stated in Equation 10, the latency ratio of RPM with respect to minimal routing is always less than 1.33 for symmetric meshes. This is significantly smaller than a penalty factor of 2 that VAL has to sacrifice to achieve optimal worst-case throughput. The latency penalty of RPM is greatly reduced for practical asymmetric topologies like the one evaluated. For the 16 × 16 × 4 configuration, using the expressions in Equations 11 and 12, the average hop count of RPM reduces to just a factor of 1.11 of DOR.

B. Detailed Flit-Level Simulation

1) Simulation Setup: The results obtained in the previous section represent upperbounds to the actual achievable throughput because it assumes an ideal single-cycle router with infinite buffers and ignores issues like flow control and contention in switches. Flit-level simulation using multi-flit packets provides more realistic insights into the performance of a routing algorithm. To achieve this, we modified the PoPnet [6] network simulator to evaluate the average routing delays under different injection loads. The PoPnet simulator models a four-stage router pipeline corresponding to routing, VC allocation, switch arbitration, and link traversal. For each simulation, we ran the simulator for 200,000 cycles. The latency of a packet is measured as the delay between the time the header flit is injected into the network and the time the tail flit is consumed at the destination. We present results for all three network configurations evaluated previously. We assume 8 virtual channels (VCs) per physical channel and buffers of size 5 flits per virtual channel. We include 8 VCs in our setup because it is well known that virtual channels improve the throughput of any routing algorithm by reducing head-of-line blocking and enabling better statistical multiplexing of flits. So, having a reasonably large number of VCs lets us compare the best performance of all routing algorithms. The injected packets are of a constant size of 5 flits. The packet size and buffer size used are the same as the parameters identified by Wang et.al. [8] as representative approximations of the on-chip networks of RAW [9] and TRIPS [10].

In addition to comparing with the oblivious routing algorithms used in the previous section, we also implemented a minimal adaptive routing algorithm based on deadlock avoidance [7] (DUATO) for comparison. The simulation was carried out for the four traffic patterns shown in Table II (Random, Transpose, Complement and DOR-WC).

<table>
<thead>
<tr>
<th>Traffic Patterns Evaluated</th>
<th>Worst-Case Throughput</th>
<th>Worst-case Traffic that causes lowest throughput.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average-Case</td>
<td>Average throughput over a million random permutations.</td>
<td></td>
</tr>
<tr>
<td>Transpose (asymmetric)</td>
<td>Packet at (x,y,z) sent to (y,z,x).</td>
<td></td>
</tr>
<tr>
<td>Complement (asymmetric)</td>
<td>Destination obtained by left shifting the concatenated bit representation of the source xyz to yzx and repartitioning the result.</td>
<td></td>
</tr>
<tr>
<td>DOR-WC</td>
<td>Packet at (x,y,z) sent to (k − z − 1,k − x − 1,k − y − 1).</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Normalized worst-case and average-case throughput</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 × 8 × 8 Network</td>
</tr>
<tr>
<td>------------------</td>
</tr>
<tr>
<td>Worst-Case</td>
</tr>
<tr>
<td>Average-Case</td>
</tr>
<tr>
<td>Transpose</td>
</tr>
<tr>
<td>Complement</td>
</tr>
<tr>
<td>DOR-WC</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>16 × 16 × 4 Network</th>
<th>Worst-Case</th>
<th>Average-Case</th>
<th>Transpose</th>
<th>Complement</th>
<th>DOR-WC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average-Case</td>
<td>0.5</td>
<td>0.125</td>
<td>0.205</td>
<td>0.25</td>
<td>0.5</td>
</tr>
<tr>
<td>Transpose</td>
<td>0.5</td>
<td>0.322</td>
<td>0.427</td>
<td>0.472</td>
<td>0.619</td>
</tr>
<tr>
<td>Complement</td>
<td>0.5</td>
<td>0.25</td>
<td>0.327</td>
<td>0.5</td>
<td>0.6</td>
</tr>
<tr>
<td>DOR-WC</td>
<td>0.5</td>
<td>0.125</td>
<td>0.214</td>
<td>0.25</td>
<td>0.5</td>
</tr>
<tr>
<td>Random</td>
<td>0.5</td>
<td>1</td>
<td>0.813</td>
<td>1</td>
<td>0.75</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>16 × 16 × 4 Network</th>
<th>Worst-Case</th>
<th>Average-Case</th>
<th>Transpose</th>
<th>Complement</th>
<th>DOR-WC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average-Case</td>
<td>0.5</td>
<td>0.083</td>
<td>0.148</td>
<td>0.25</td>
<td>0.5</td>
</tr>
<tr>
<td>Transpose</td>
<td>0.5</td>
<td>0.25</td>
<td>0.367</td>
<td>0.286</td>
<td>0.5</td>
</tr>
<tr>
<td>Complement</td>
<td>0.5</td>
<td>0.5</td>
<td>0.196</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>DOR-WC</td>
<td>0.5</td>
<td>0.083</td>
<td>0.218</td>
<td>0.333</td>
<td>0.667</td>
</tr>
<tr>
<td>Random</td>
<td>0.5</td>
<td>1</td>
<td>0.758</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
2) Simulation Results: The simulation results are presented in Figures 3, 4 and 5. The results follow a trend similar to the simplified throughput analysis presented in the previous section. RPM consistently achieves good throughput over all traffic patterns considered. The saturation throughput of RPM is higher than VAL and its latency is strictly lower over all traffic patterns considered. The saturation throughput is primarily bounded by the saturation throughput of the horizontal channels. For this topology, RPM sustains the highest (or very close to highest) throughput for all four topologies considered.

Although DOR and O1TURN perform well on uniform and bit-complement traffic, their performance degrades significantly when subjected to a worst-case traffic pattern for DOR (DOR-WC), where both are clearly outperformed by RPM. The normalized throughput of O1TURN and DOR for the DOR-WC traffic pattern degrades with the increase in network radix from 4 to 8. On the other hand, the normalized throughput of RPM changes very little when the network radix is increased.

RPM also outperforms ROMM and DUATO for all traffic patterns considered except uniform traffic on the symmetric mesh topologies. For the asymmetric mesh, RPM is comparable to DOR and O1TURN and better than ROMM and DUATO for uniform traffic. This is because, for the \(16 \times 16\) mesh, two phase routing on the short vertical dimension no longer forms a throughput bottleneck and the overall throughput is primarily bounded by the saturation throughput of the horizontal channels. For this topology, RPM sustains the highest (or very close to highest) throughput for all four traffic patterns evaluated. The poor performance of ROMM and DUATO, especially on bit-complement traffic, despite having sufficient path diversity can be attributed to the fact that they are restricted to routing in the minimal cube. This results in congestion of links in the middle of the network. RPM achieves better load balancing by using non-minimal paths.

Lastly, the latency of RPM is slightly higher than the minimal routing algorithms when the network is lightly loaded. The latency difference can be clearly seen for the \(8 \times 8 \times 8\) topology. However, the difference in latency is much less when compared to VAL and reduces significantly for the asymmetric configuration.

The results clearly validate the claim that O1TURN, which achieves near-optimal worst case throughput for 2D meshes,
performs poorly in the worst-case sense when extended to 3D. RPM, on the other hand handles adversarial traffic much better than any of the other minimal routing algorithms considered (oblivious or adaptive). It does so while paying a far smaller latency penalty compared to VAL.

VI. CONCLUSION

The increasing viability of three dimensional silicon integration technology has opened new opportunities for chip architecture innovations. The main contribution of this paper is the design of a new oblivious routing algorithm for 3D mesh networks called Randomized Partially-Minimal (RPM) routing. Mesh networks constitute an important class of interconnection networks that matches well with tile based microarchitectures. Although minimal routing with near-optimal worst-case throughput has already been achieved for the 2D case using an algorithm called O1TURN [4], the optimality of O1TURN does not extend to 3D or higher dimensions. RPM provably achieves optimal worst-case throughput for 3D meshes when the network radix $k$ is even and within a factor of $1/k^2$ of optimal worst-case throughput when $k$ is odd. RPM also outperforms VAL, DOR, ROMM, and O1TURN in average-case throughput by 33.3%, 111%, 47%, and 30%, respectively on an $8 \times 8 \times 8$ mesh. Finally, whereas Valiant’s routing algorithm (VAL) [2] achieves optimal worst-case throughput at a penalty factor of 2 in average latency over DOR, RPM achieves (near) optimal worst-case throughput with a much smaller factor of 1.33. In practice, the average latency of RPM is expected to be closer to minimal routing because 3D mesh networks are not expected to be symmetric in 3D chip designs. The number of available device layers is expected to be much less than the number of processor tiles that can be placed along an edge of a device layer. For practical asymmetric 3D mesh configurations, the average latency of RPM reduces to just a factor of 1.11 of DOR.

REFERENCES