

# A Layer-Multiplexed 3D On-Chip Network Architecture

Rohit Sunkam Ramanujam and Bill Lin

Electrical and Computer Engineering Department, University of California, San Diego

**Abstract**—Programmable many-core processors are poised to become a major design option for many embedded applications. In the design of power-efficient embedded many-core processors, the architecture of the on-chip network plays a central role. Many designs have relied on a 2D mesh architecture as the underlying communication fabric. With the emergence of 3D technology, new on-chip network architectures are possible. In this paper, we propose a novel layer-multiplexed (LM) 3D network architecture that takes advantage of the short inter-layer wiring delays enabled in 3D technology. In particular, the LM architecture replaces the one-layer-per-hop routing in a conventional 3D mesh with simpler vertical demultiplexing and multiplexing structures. When combined with a layer-multiplexing oblivious routing algorithm, it can achieve the same worst-case throughput as the best known oblivious routing algorithm on a conventional 3D mesh. However, in comparison to a conventional 3D mesh, the LM architecture consumes 27% less power, attains 14.5% higher average throughput, and achieves 33% lower worst-case hop count.

## I. INTRODUCTION

RECENT advances in power-efficient many-core processor architectures [2]–[5] have made it possible to achieve a huge amount of processing performance per watt without the long expensive development cycles of custom ASIC designs. Thus, they have become poised for significant adoption in many embedded applications where performance and power-efficiency are both important. With the emergence of viable three-dimensional (3D) integration technology [6], [7], opportunities exist for new chip architecture innovations. Indeed, 3D integration has attracted significant attention in recent years because of its potential benefits, including smaller chip footprints, higher transistor density, shorter wiring delays, and significantly higher communication bandwidth. One direction is to extend existing 2D tile-based many-core processor architectures [1]–[5] into three dimensions [8], [9]. Many proposed 2D tile-based processor architectures have relied on a 2D mesh topology as the underlying network fabric. Therefore, extending mesh-based tiled architectures into three dimensions represents a natural progression.

As with 2D mesh networks, throughput and latency are important performance metrics in the design of routing algorithms for 3D meshes. Recently, a near-optimal oblivious routing algorithm for conventional 3D mesh networks called Randomized Partially-Minimal (RPM) routing was proposed [10]. Fig. 1 depicts how RPM works. Suppose  $Z$  is the vertical dimension and  $X$  and  $Y$  are the two horizontal dimensions. RPM first routes in the  $Z$  dimension to a randomly chosen intermediate  $XY$  plane. It then routes flits on each  $XY$  plane using either minimal  $XY$  or  $YX$  routing with equal probability. Finally, it routes flits to their final destinations along the  $Z$  dimension. Essentially, RPM works by load-balancing traffic uniformly across vertical layers and routing minimally on each horizontal layer, using either a  $Z$ - $XY$ - $Z$  or  $Z$ - $YX$ - $Z$  path. The main result shown in [10] is that RPM achieves optimal worst-case throughput when the network radix  $k$  is even and within a factor of  $1/k^2$  of optimal when  $k$  is odd.

For a symmetric 3D mesh, RPM achieves a factor of 1.33x of dimension-ordered routing (DOR) [11] in average hop count. However, it is well-known that DOR suffers from poor worst-case and average-case throughput due to a lack of path diversity. The best previously known oblivious routing algorithm that achieves optimal worst-case throughput for 3D meshes is Valiant (VAL) routing [12], which globally load-balances across all dimensions

to achieve optimal worst-case throughput, but at the expense of destroying locality. VAL is a factor of 2x of DOR in average hop count. Although O1TURN [13] has been shown to achieve both near-optimal worst-case throughput and minimal hop count for 2D meshes, its performance surprisingly degrades tremendously for the 3D case, as noted in [13]. ROMM [14] provides another alternative minimal routing algorithm, but it too suffers from poor worst-case throughput. Fig. 2 shows the worst-case throughput of RPM in comparison to VAL, DOR, ROMM and O1TURN. The worst-case throughput of DOR, ROMM and O1TURN 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 ROMM and O1TURN.

In this paper, we present a novel layer-multiplexed (LM) architecture for 3D interconnection networks that exploits the optimality of RPM together with the short inter-layer wiring delays enabled in 3D technology. In contrast to a conventional 3D mesh, the LM architecture replaces the *one-layer-per-hop* routing in the vertical dimension with simpler vertical *demultiplexing* and *multiplexing* structures. In doing so, the LM architecture retains the same worst-case throughput as a conventional 3D mesh by adapting RPM routing to the LM architecture. However, the LM architecture consumes 27% less power, attains 14.5% higher average throughput, and achieves 33% lower worst-case hop count.

## II. RELATED WORK

Li et al. [9] proposed a NOC-Bus-hybrid architecture, which uses a shared bus for inter-layer communication. However, under adversarial traffic conditions, e.g. when the source and destinations are on different layers, the shared bus becomes the throughput bottleneck. Kim et al. [15] proposed a true 3D crossbar design called DimDe. As noted in [15], its design can only support routing using DOR, which is known to suffer from poor worst-case throughput. Matsutani et al. [16] proposed an architecture called XNOTs which is also based on the idea of vertical switching. However, it requires large  $2k \times 2k$  vertical switches, which are costly since crossbar power grows quadratically with the number of ports. Park et al. [17] proposed MIRA, which is based on implementing a 2D mesh chip-multiprocessor in three dimensions. It assumes the processor cores are designed in 3D, which makes it difficult to reuse existing highly optimized 2D processor core designs.

## III. LAYER-MULTIPLEXED 3D ARCHITECTURE

A 3D mesh is a natural extension to the 2D mesh architecture. Two extra ports are needed at each router in the 3D case to support up/down communication, resulting in a  $7 \times 7$  crossbar, as compared to a  $5 \times 5$  crossbar in the 2D case. However, since the crossbar power cost increases quadratically with the number of ports, the power consumption of 3D routers is much higher than the 2D case. Also, in a 3D mesh, packets are routed in a *one-layer-per-hop* manner along the vertical dimension. When RPM routing is used, which involves two phases of vertical routing, a packet may need to traverse  $2k$  layers (and hence  $2k$  router hops) in the worst-case, resulting in long worst-case latencies. This poor worst-case performance does not leverage the short inter-layer distances in 3D designs. This motivated us to propose a new Layer-Multiplexed (LM) 3D on-chip network architecture that exploits the optimality of RPM together with the

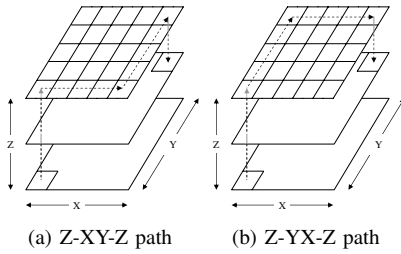


Fig. 1. Examples of RPM routing

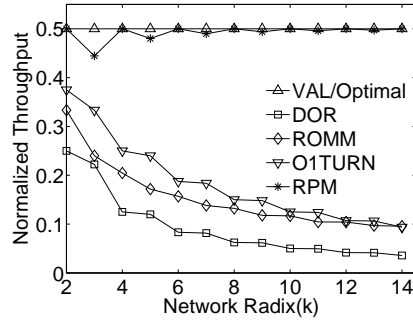


Fig. 2. Normalized 3D worst-case throughput

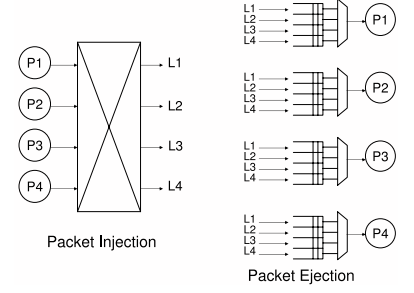


Fig. 3. Injection and Ejection Stages

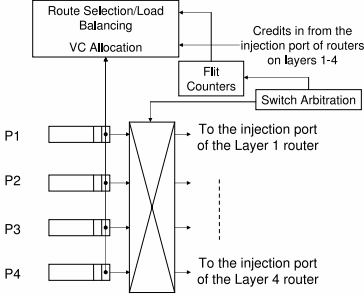


Fig. 4. Packet injection microarchitecture

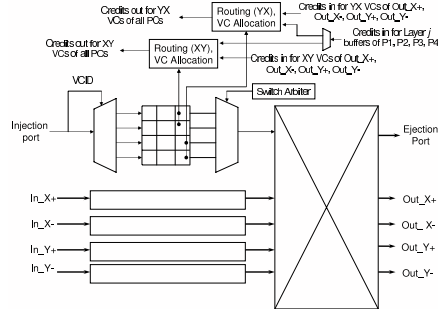


Fig. 5. Intra-layer router microarchitecture

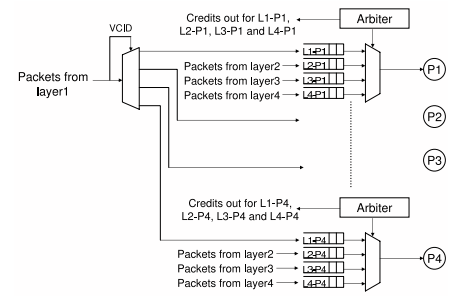


Fig. 6. Packet ejection microarchitecture

short inter-layer wiring delays and the abundance of vertical wiring in 3D designs. The LM architecture has a lower power cost and higher performance compared to a 3D mesh.

#### A. Architecture

In the LM architecture, the one-layer-per-hop communication in a 3D mesh is replaced with simpler demultiplexing and multiplexing stages. This is shown in Fig. 3 for  $k = 4$  layers. These demultiplexing and multiplexing stages are used to implement the layer load-balancing method employed by RPM. At packet injection, flits are *demultiplexed* uniformly to the  $k$  layers using the *packet injection stage*. Once demultiplexed to a horizontal layer, flits are routed on this plane using either minimal XY or YX routing with equal probability. At the destination (X, Y) coordinates, packets from all layers are *multiplexed* at the destination processor in the *packet ejection stage*.

With this architecture, each horizontal plane is effectively a conventional 2D mesh with just 5-port routers. Rather than connecting directly to these 5-port routers, each processor is connected to a packet injection stage at the same (X, Y) location. This packet injection stage is in turn connected to  $k$  5-port routers at the same (X, Y) location. This is depicted in Fig. 3 for  $k = 4$ , with processors  $P_1$  to  $P_4$  demultiplexing to layers  $L_1$  to  $L_4$ . Similarly, on the packet ejection side, a packet ejection multiplexer is used at each processor to multiplex flits arriving from layers  $L_1$  to  $L_4$ . We refer to this adaptation of RPM routing on the LM architecture as RPM-LM.

As we will see in Section IV-A, the combined costs of these demultiplexers and multiplexers with the 5-port router costs are significantly lower than the 7-port router costs of a conventional 3D mesh. The microarchitectural details of these packet injection and ejection stages and the routers are detailed next.

1) *Packet Injection*: The packet injection stage is essentially a 4-port switch with a typical router pipeline consisting of route selection, VC allocation, switch arbitration, switch traversal, and link traversal. However, route selection is modified to implement layer load-balancing, which has to ensure that statistically traffic does indeed get uniformly distributed across all  $k$  layers. This is achieved by using a set of flit-counters for each ordered pair of input and output

ports in the load-balancing logic. A flit-counter  $(i, j)$  records the total number of flits sent from input  $i$  to output  $j$ . When a new head flit is injected from processor  $P_i$ , the route selection logic will select the output layer  $L_j$  with the lowest flit-count  $(i, j)$ . Ties between layers are broken by a rotating output priority pointer for  $P_i$  that gets advanced at each route selection.

Once a layer is chosen by the route selection stage for the head flit, all remaining flits of the same packet will be routed on the same layer. The VC allocation stage further allocates a virtual channel on the injection port of the 5-port router on the selected layer. Together, flits belonging to the same packet are guaranteed to be routed on the same layer through the same set of virtual channels, thus ensuring their in-order arrival at the destination.

Finally, we found that a single VC with a small amount of buffering (e.g. 5 flits) is sufficient at each input of the packet injection demultiplexing switch. For implementation, the packet injection demultiplexer can be spread across the  $k$  layers. The control logic can be located on the middle layer so that the wiring for the control signals is minimized and uniformly distributed across the layers.

2) *Router Microarchitecture*: Fig. 5 depicts the microarchitectural details of the 5-port horizontal plane routers used in the LM architecture. Once a packet is injected into a router on one of the layers, RPM-LM uses either minimal XY or YX paths with equal probability to reach the (X, Y) coordinates of its destination. This is essentially O1TURN [13] routing on the horizontal plane. In turn, the microarchitecture of each 5-port router on the 2D plane is very similar to that of the O1TURN router [13]. The injection port receives flits from any one of the outputs of the vertical switch. The virtual channels at each input port are divided into two sets—one set for XY routing and another for YX routing. After being injected into one of the routing layers, a packet is restricted to remain in the virtual channels of that layer while being routed on the 2D plane. The routing and VC allocation stages are duplicated as in the O1TURN router to independently handle the corresponding decisions in the XY and YX routings. The switch arbitration stage is common to both XY and YX routings since flits from all virtual channels at each input contend for

the same crossbar.

3) *Packet Ejection*: The ejection port of each router is connected through packet ejection multiplexers to all processors at the same  $(X, Y)$  coordinate, as depicted in Fig. 6. In particular, each horizontal plane router sees at its ejection port four virtual channels, each of which corresponds to an ejection queue of a processor connected to the router. In the example depicted in Fig. 6, a router on the first layer ( $L_1$ ) sees four virtual channels, namely  $L_1-P_1$ ,  $L_1-P_2$ ,  $L_1-P_3$ , and  $L_1-P_4$  as its ejection channels located at processors  $P_1$ ,  $P_2$ ,  $P_3$  and  $P_4$ , respectively. The route selection stage chooses the appropriate output VC for a packet based on its destination. After leaving the horizontal plane router, the VC ID field on a flit is used to determine the packet ejection multiplexer into which the flit is inserted. Finally, each packet ejection multiplexer can independently choose among its input queues which flit to forward to the corresponding destination processor at each cycle. Each multiplexer also handles flow control for its input buffers. We found that only a small amount of buffering (e.g. 5 flits) is sufficient at each of these input queues. For implementation, each packet ejection multiplexer can be implemented on the corresponding processor layer.

#### B. Worst-Case Throughput Analysis

*Claim 1*: The worst-case throughput of RPM-LM is equal to the worst-case throughput of RPM on a 3D mesh, which is optimal when the network radix  $k$  is even and within a factor of  $1/k^2$  of optimal when  $k$  is odd.

*Proof*: As in [10], optimal worst-case throughput analysis is based on ideal load analysis, which assumes ideal routers with infinite buffering. It is well-known that Valiant routing (VAL) [12] achieves optimal worst-case throughput, and its maximum channel load is twice that of uniform traffic under dimension-ordered routing, which is  $k/2$  when  $k$  is even and  $(k^2-1)/2k$  when  $k$  is odd. To demonstrate that a routing algorithm  $R$  is worst-case throughput optimal, it is sufficient to show that the maximum channel load of  $R$  under all admissible traffic is the same as VAL. In [10], it was shown that RPM on a 3D mesh is optimal when  $k$  is even and within a factor of  $1/k^2$  of optimal when  $k$  is odd. This was shown by showing that the uniform load-balancing of RPM in the vertical dimension ensures that the 2D traffic that will traverse any XY plane will be doubly sub-stochastic if the overall 3D traffic matrix  $\Lambda$  is doubly sub-stochastic. Given that RPM uses O1TURN [13] for routing on a XY plane, and that the 2D traffic on a XY plane is guaranteed to be admissible, it follows from the O1TURN result that the maximum channel load along the horizontal dimensions is  $k/2$ , which is optimal when  $k$  is even. When  $k$  is odd, the ratio of the maximum channel loads of VAL over RPM is  $[(k^2-1)/2k]/[k/2] = (1-1/k^2)$ , which is a factor  $1/k^2$  of optimal.

For RPM-LM routing on the LM architecture, traffic is also uniformly load-balanced across the layers. Using the same analysis, the 2D traffic that will traverse any XY plane will be doubly sub-stochastic under an admissible  $\Lambda$ . Therefore, the maximum channel load on the XY planes will be the same as O1TURN. Given that the demultiplexing packet injection stage and the multiplexing packet ejection stage are both non-blocking under ideal load analysis, the maximum channel load is dictated by the load on the channels in the XY planes. Therefore, RPM-LM on the LM architecture achieves the same worst-case throughput as RPM on a 3D mesh. This analysis holds for both symmetric and asymmetric meshes. ■

## IV. EVALUATION

In this section, we evaluate the LM architecture in comparison to a conventional 3D mesh in terms of power and performance. We evaluate two 3D mesh configurations, a  $4 \times 4 \times 4$  symmetric mesh and

a  $8 \times 8 \times 4$  asymmetric mesh topology. Both the 3D configurations evaluated have four device layers. In practice, 3D mesh networks are not expected to be symmetric in 3D chip designs. The number of device layers is expected to be much less than the number processor tiles that can be placed along the edge of a layer. Hence, we chose an asymmetric  $8 \times 8 \times 4$  mesh topology for our evaluation.

TABLE I  
POWER ANALYSIS

	Router Power (mw)	Amortized Demux Power (mW)	Amortized Mux Power (mW)	Total Avg. Router Power (mW)
3D mesh	270.2	n.a.	n.a.	270.2
LM	153.66	22.56	21.05	197.27

#### A. Power Comparison

Table I compares the average router power cost per processor for the two architectures. This comparison holds for both the symmetric and the asymmetric topologies which have the same number of device layers. The routers in the LM architecture have only 5 ports, compared to 7 ports needed by routers in a 3D mesh. However, the LM architecture incurs the cost of an extra vertical demultiplexing switch for every 4 routers and a buffered multiplexer at every processor. The additional hardware adds to the amortized router cost per tile. We use the power models in Orion [20] for our power analysis. The estimated power includes both dynamic and leakage power components. The power models used are for a 65 nm process technology with the supply voltage set at 1V. We assume 4 GHz operating frequency and 30% activity at each input port. For each router, we assume input buffers consisting of 4 virtual channels (VC), each 5 flits deep in both architectures. The vertical demultiplexing switch in the LM architecture has a single queue with 5 flits of buffering at each input. We found that a small amount of buffering at the demultiplexer is sufficient to load-balance flits across the vertical layers. Each multiplexer at a processor in the LM architecture has 4 queues, one for each layer, with 5 flits of buffering each. As shown in Table I, with the costs of the demultiplexing and multiplexing structures added, the LM architecture still consumes 27% less power in comparison to a 3D mesh.

TABLE II  
TRAFFIC PATTERNS EVALUATED

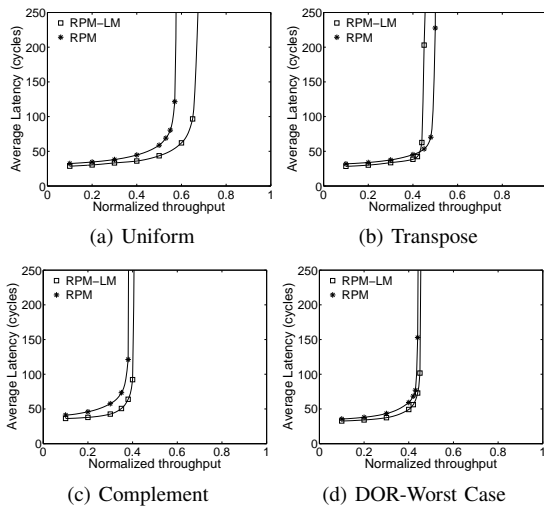
Worst-Case	Worst-case traffic that causes lowest throughput.
Average-Case	Average throughput over a million random matrices.
Transpose	Packet at $(x, y, z)$ sent to $(y, z, x)$ .
Complement	Packet at $(x, y, z)$ sent to $(k_x - x - 1, k_y - y - 1, k_z - z - 1)$ .
DOR-WC	Packet at $(x, y, z)$ sent to $(k - z - 1, k - y - 1, k - x - 1)$ .
Uniform	Packet sent to destination at uniform random.

TABLE III  
RPM vs. RPM-LM

	$4 \times 4 \times 4$		$8 \times 8 \times 4$	
	3D mesh	LM	3D mesh	LM
Worst-Case	0.5	0.5	0.5	0.5
Average-Case	0.62	0.71	0.7254	0.73
Transpose	0.6	0.53	0.5	0.5
Complement	0.5	0.5	0.5	0.5
DOR-WC	0.5	0.5	0.5	0.5
Uniform	0.75	1	1	1

#### B. Throughput Evaluation

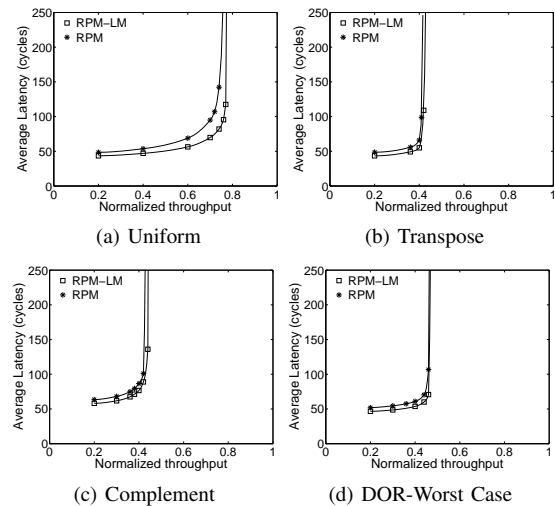
In this section, we compare the throughput of RPM-LM on the LM architecture with RPM on a 3D mesh. These routing algorithms have been shown to be worst-case throughput optimal for the two architectures, and RPM is known to be better than other existing algorithms for 3D meshes, especially under adversarial traffic. Throughput is

Fig. 7. Performance of RPM on 3D mesh and LM for the  $4 \times 4 \times 4$  case

evaluated in two parts. First, we assume ideal single-cycle routers with infinite buffers and perform a simplified throughput analysis. We then back these results with detailed flit-level simulations.

1) *Simplified Throughput Analysis:* Table III presents simplified throughput analysis results. The throughput values are normalized to the network capacity of a 3D mesh, where the network capacity is defined by the maximum channel load that a channel at the bisection of a network needs to sustain under uniform traffic. The worst-case traffic for RPM and RPM-LM is determined by the method proposed in [18]. The average-case throughput is computed by averaging the throughput over a large set of randomly generated traffic patterns. For the average-case results shown in Table III, we averaged over one million random traffic patterns. We also compare the throughput of the two architectures on the four different traffic patterns defined in Table II. RPM-LM has the same worst-case throughput as that of RPM on a 3D mesh, as analyzed in Section III-B. For the symmetric  $4 \times 4 \times 4$  mesh topology, RPM-LM outperforms RPM by 14.5% in average-case throughput, which is a significant improvement. For this topology, although RPM performs slightly better than RPM-LM on transpose traffic, the two perform comparably on Complement and DOR-WC traffic patterns. On uniform traffic, RPM-LM outperforms RPM by 33% for the symmetric mesh topology. This is mainly because the LM architecture makes better use of the inter-layer bandwidth available in 3D ICs to overcome bottlenecks caused by two-phase routing along the vertical dimension. The simplified throughput analysis results of RPM and RPM-LM are very similar for the asymmetric  $8 \times 8 \times 4$  mesh topology. RPM-LM and RPM have the same worst-case throughput and comparable average-case throughput. The saturation throughputs for the traffic patterns evaluated are also very close. This is because, for the asymmetric topology, throughput is mainly constrained by the load on the horizontal channels and the links along the short vertical dimension are relatively less loaded.

2) *Detailed Flit-Level Simulation:* We use flit-level simulations to compare the actual throughput that can be sustained by the two architectures under different traffic conditions. In particular, we used the PoPnet [19] simulator to evaluate the average routing delays under different injection loads. The simulator models a five stage router pipeline corresponding to route selection, VC allocation, switch arbitration, switch traversal, and link traversal. Each input has 8 VCs with 5 flit buffers per VC. 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. The injected packets have a constant size of 5 flits. The simulations were performed on

Fig. 8. Performance of RPM on 3D mesh and LM for the  $8 \times 8 \times 4$  case

the four traffic patterns shown in Table II. As shown in Fig. 7, these results follow the same trend as the simplified throughput analysis.

### C. Worst Case Hop Count Analysis

Finally, the worst-case hop count of RPM is given as  $(k_x - 1) + (k_y - 1) + 2(k_z - 1)$  and the worst-case hop count of RPM-LM is given as  $(k_x - 1) + (k_y - 1) + 2$  where  $k_x$ ,  $k_y$ ,  $k_z$  are the lengths of the  $X$ ,  $Y$  and  $Z$  dimensions, respectively and the “+2” corresponds to the extra demultiplexing and multiplexing hops.. The LM architecture achieves a worst-case hop count reduction of 33% for the symmetric topology and 20% for the asymmetric topology.

### REFERENCES

- [1] M. B. Taylor et al., “Scalar operand networks: On-chip interconnect for ILP in partitioned architectures,” *HPCA*, 2003.
- [2] S. Vangal et al., “An 80-tile sub-100-W teraFLOPS processor in 65nm CMOS,” *IEEE Journal of Solid State Circuits*, Jan 2008.
- [3] A. Agarwal et al., “Tile Processor: Embedded Multicore for Networking and Multimedia,” *Hot Chips*, Aug 2007.
- [4] R. Wilson, “Cisco taps processor array architecture for NPU,” *EE Times*, August 9, 2004
- [5] <http://www.picochip.com>
- [6] K. Lee et al., “Three-dimensional shared memory fabricated using wafer stacking technology,” *IEDM Technical Digest*, Dec 2000.
- [7] W. R. Davis et al., “Demystifying 3D ICs: The pros and cons of going vertical,” *IEEE Design & Test of Computers*, 2005.
- [8] T. Kgil et al., “PICOSERVER: Using 3D stacking technology to enable a compact energy efficient chip multiprocessor,” *ASPLOS*, 2006.
- [9] F. Li et al., “Design and management of 3D chip multiprocessors using network-in-memory,” *ISCA*, 2006.
- [10] R. S. Ramanujam, B. Lin, “Near-optimal oblivious routing on three-dimensional mesh networks,” *ICCD*, 2008.
- [11] H. Sullivan, T. R. Bashkow, “A large scale, homogeneous, fully distributed parallel machine,” *Annual Symp. on Comp. Arch.*, 1977.
- [12] L. G. Valiant, G. J. Brebner, “Universal schemes for parallel communication,” *ACM STOC*, 1981.
- [13] D. Seo et al., “Near-optimal worst-case throughput routing for two-dimensional mesh networks,” *ISCA*, 2005.
- [14] T. Nesson, S. L. Johnsson, “ROMM routing on mesh and torus networks,” *ACM SPAA*, 1995.
- [15] J. Kim et al., “A novel dimensionally-decomposed router for on-chip communication in 3D architectures,” *ISCA*, 2007.
- [16] H. Matsutani et al., “Tightly-Coupled Multi-Layer Topologies for 3-D NOCs,” *International Conference on Parallel Processing*, 2007.
- [17] D. Park et al., “MIRA: A Multi-layered On-Chip Interconnect Router Architecture,” *ISCA*, 2008.
- [18] B. Towles, W. J. Dally, “Worst-case traffic for oblivious routing functions,” *ACM SPAA*, 2002.
- [19] L. Shang et al., “Dynamic voltage scaling with links for power optimization of interconnection networks,” *HPCA*, 2003.
- [20] H. Wang, X. Zhu, L.-S. Peh, S. Malik, “Orion-A power-performance simulator for interconnection networks,” *MICRO-35* 2002.