Appears in the Proceedings of the 15^{th} International Symposium on High-Performance Computer Architecture

Express Cube Topologies for On-Chip Interconnects

Boris Grot Joel Hestness Stephen W. Keckler Onur Mutlu[†]

Department of Computer Sciences
The University of Texas at Austin
{bgrot, hestness, skeckler}@cs.utexas.edu

[†]Computer Architecture Laboratory (CALCM)
Carnegie Mellon University
onur@cmu.edu

Abstract

Driven by continuing scaling of Moore's law, chip multiprocessors and systems-on-a-chip are expected to grow the core count from dozens today to hundreds in the near future. Scalability of on-chip interconnect topologies is critical to meeting these demands. In this work, we seek to develop a better understanding of how network topologies scale with regard to cost, performance, and energy considering the advantages and limitations afforded on a die. Our contributions are three-fold. First, we propose a new topology, called Multidrop Express Channels (MECS), that uses a one-to-many communication model enabling a high degree of connectivity in a bandwidth-efficient manner. In a 64-terminal network, MECS enjoys a 9% latency advantage over other topologies at low network loads, which extends to over 20% in a 256-terminal network. Second, we demonstrate that partitioning the available wires among multiple networks and channels enables new opportunities for trading-off performance, area, and energy-efficiency that depend on the partitioning scheme. Third, we introduce Generalized Express Cubes – a framework for expressing the space of on-chip interconnects – and demonstrate how existing and proposed topologies can be mapped to it.

1 Introduction

As continuing scaling of Moore's law enables ever greater transistor densities, design complexity and power limitations of conventional out-of-order superscalar processors have forced researchers to consider new applications for large transistor budgets of today and tomorrow. Single-chip multiprocessors, or CMPs, have emerged as the leading alternative to complex monolithic uniprocessors. By placing multiple cores on a single die, complexity is managed via replication and design reuse, while power is kept in check through the use of less aggressive microarchitectures. Today's most expansive designs have dozens of tiled cores on a chip and include the 64-core Tile Processor from Tilera [25] and Intel's 80-core Terascale chip [22]. Continuing technology scaling will likely push single silicon substrates to integrate in the near future.

To interconnect such a high number of elements on a die, researchers and designers have turned to interconnection networks as a replacement to conventional shared buses and ad-hoc wiring solutions. Most existing networks-on-chip (NOCs) are based on rings [16] or two-dimensional meshes [25, 22, 9, 23] – topologies that have low design complexity and are a good fit to planar silicon substrates.

These topologies, however, present serious scalability challenges as the core count increases toward hundreds or thousands, especially as two-dimensional substrates restrict the space of implementable networks. In response, researchers have proposed *concentration* as a means to reduce the number of network nodes by co-locating multiple terminals at each network interface with a crossbar interconnect [1]. Another solution involves *flattening* a conventional butterfly network for use on a chip with the aim of reducing the hop count through the use of high-radix routers [10]. Unfortunately, neither approach is sufficiently scalable. By itself, concentration is insufficient as its degree is restricted by crossbar complexity, while the flattened butterfly requires channel count that is quadratic in the number of interconnected nodes.

To address these scalability concerns, we introduce Multidrop Express Channels (MECS) – a new topology based on express cubes [4] that is specifically designed to fit the unique advantages and constraints of NOCs. MECS utilize a point-to-multipoint communication fabric that provides a high degree of connectivity in a bandwidth-efficient manner. We use an analytical model to understand how MECS, concentrated mesh, and flattened butterfly topologies scale when the network size is increased from 64 to 256 terminals. As the network is scaled, MECS maintains a low network diameter and requires only a linear increase in bisection bandwidth to keep the channel width constant. An evaluation of MECS on a subset of the PARSEC benchmark suite in a 64-terminal system shows that MECS enjoys a latency advantage exceeding 9% over other topologies, including the flattened butterfly. Scalability studies with synthetic benchmarks show that the latency benefit of MECS increases to at least 20% at low loads in a 256-terminal configuration.

To better understand the space of on-chip interconnects, we propose Generalized Express Cubes (GEC) – a framework that extends k-ary n-cubes with concentration and ex-

press channels – and demonstrate how various topologies, including MECS and flattened butterfly, can be expressed in it.

Finally, we evaluate several GEC-expressible networks that differ in channel count, connectivity and bandwidth. Our findings show that in wire-rich substrates, completely replicating the networks while holding the bisection bandwidth constant can significantly improve network throughput at a modest delay penalty at low loads. In addition, replication can reduce router area and power by decreasing the crossbar complexity.

2 Background

In this effort, we seek to determine how interconnect topologies scale with respect to cost, performance, and energy-efficiency. The study of other relevant factors, such as fault-tolerance, is left as future work.

2.1 Performance

The performance of interconnection networks is determined by two factors: throughput and latency [5]. Throughput is the maximum rate at which the network can accept the data. Latency is the time taken by the packet to traverse the network from the source to the destination and is comprised of two components: header latency, T_h, and serialization delay, T_s.

As shown in Equation 1, the header latency is the sum of router delay, d_r, and wire delay, d_w, at each hop, multiplied by the hop count, H. The serialization latency (Equation 2) is the number of cycles required by a packet to cross the channel and is simply the quotient of the packet length, L, and the channel width, W. The resulting expression in Equation 3 is known as the *zero-load* latency. In practice, contention between different packets in the network can increase the router and/or serialization delay, leading to higher packet latencies. A good topology seeks to minimize network latency and maximize throughput.

$$T_h = (d_r + d_w)H \quad (1)$$

$$T_s = L/W \quad (2)$$

$$T = T_h + T_s = (d_r + d_w)H + L/W \quad (3)$$

By far, the most popular NOC topology to date has been a two-dimensional mesh [25, 22, 9, 23]. Given the short channel lengths in on-chip meshes, the typical per-hop wire delay in these networks is one cycle. Since aggressively-clocked implementations require pipelined routers, researchers have turned to techniques like speculation [15, 12] and express virtual channels [11] to reduce the router latency to one or two cycles per hop. However, because meshes require a large number of hops, particularly as the network size scales, router latency in two-dimensional meshes remains a major component of network latency.

2.2 Area

Traditionally, the cost of interconnection networks has been dictated primarily by pin constraints of the available packaging technology. In networks-on-chip (NOCs), however, die area and wiring complexity are the main determinants of network cost. The area overhead is due to routers and communication channels. Link overhead is mostly dominated by the area of repeaters, as wires are commonly routed in higher-level metal layers. Equation 4 approximates channel footprint as a product of area cost per mm of wire, channel width, and combined channel span in millimeters.

Flit buffers, crossbars, and control logic are the primary contributors to the routers' area cost (Equation 5). However, since control logic has a negligible footprint [8, 9], we will restrict our subsequent analysis to crossbar and buffer overheads.

$$A_{links} = A_{wire_{mm}} \cdot w \cdot \sum_{i=1}^{C} \delta_i \quad (4)$$

$$A_{routers} = (A_{fifo} + A_{crossbar} + A_{arbiters}) \cdot N \quad (5)$$

$$A_{NOC} = A_{links} + A_{routers} \quad (6)$$

The wiring complexity, combined with restrictions imposed by planar silicon substrates, profoundly affects the choice of topologies suitable for networks on a chip. Simple, low-dimensional topologies such as rings and two-dimensional meshes are appealing for use in on-chip networks as they are straightforward to implement in silicon, have short channel lengths, and have low router complexity. High-dimensional k-ary n-cube and k-ary n-fly topologies require flattening for mapping to a 2D substrate. This can limit their scalability and introduce non-minimal channel lengths, which tend to adversely impact wire delay and energy, and can complicate the routability of the design.

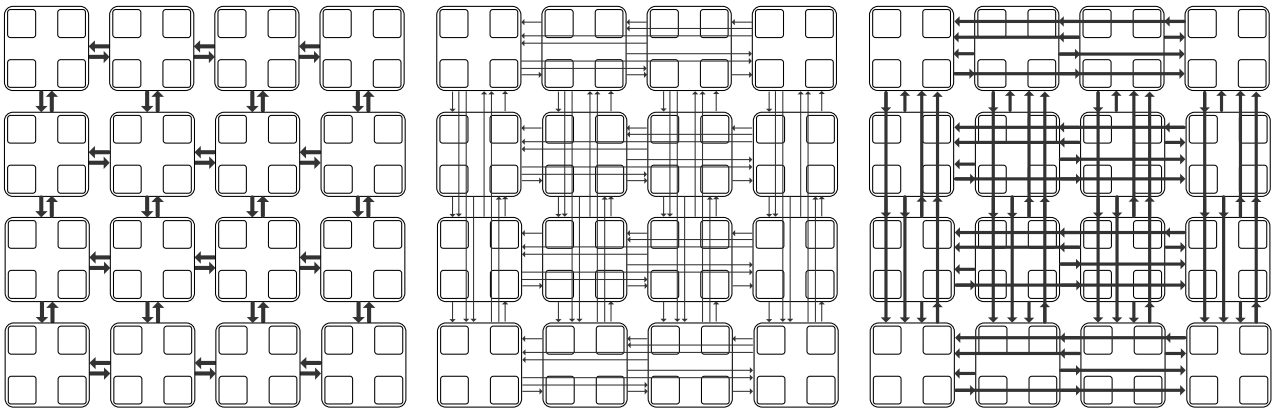
2.3 Energy

Links and routers are the two primary contributors to NOC energy. In point-to-point on-chip interconnects, repeaters are inserted at regular intervals to improve the channel's energy and delay characteristics. As a result, link energy is proportional to the wire distance and can be approximated as the product of the energy per millimeter of wire, the channel width *w*, and the sum of all of the per-hop wire lengths (δ_i) in millimeters between the source and the destination (Equation 7). The summation allows for the non-uniform hop distances found in some topologies.

$$E_{link} = E_{wire_{mm}} \cdot w \cdot \sum_{i=1}^{H} \delta_i \quad (7)$$

$$E_{router} = \sum_{i=1}^{H} (E_{fifo} + E_{crossbar} + E_{arbiters}) \quad (8)$$

$$E_{NOC} = E_{link} + E_{router} \quad (9)$$



(a) Concentrated mesh (b) Flattened butterfly (c) MECS

Figure 1. Concentrated Mesh, Flattened Butterfly and MECS topologies for a 64-terminal network.

The main contributors to a router's energy footprint are the flit FIFOs, the internal switching logic, and the arbiters. FIFO energy depends on the number of virtual channels per router port as well as the depth and width of each virtual channel (VC). On-chip routers are typically designed with a crossbar switch. In crossbars with an equal number of input and output ports, energy per flit can be approximated as $n{\cdot}w^2$, where n is the number of switch ports and w is the width of each port. Finally, arbiters typically contribute a small fraction to router's energy overhead and are included in our analysis for completeness.

As shown in Equation 8, the combined router energy must be scaled by the hop count to yield the full contribution of router energies to the total network energy. Equation 9 is simply the sum of router and channel energy values and represents the total energy required to deliver a single flit. Under this simplified model, distinct topologies that route a packet over the same Manhattan distance would incur the same link energy cost but dissimilar router energy as a result of differences in router microarchitecture and hop count.

2.4 Scalability

Since today's most aggressive CMP designs have dozens of cores, CMPs featuring hundreds or thousands of processing elements are likely to enter the mainstream in the near future. Scalable on-chip interconnect fabrics will be critical to the success of these systems and must attain high performance with low cost and sufficient energy efficiency.

Cost-effective, simple rings appear to be the least scalable alternative, since the hop count – and thus, latency and energy – grows linearly with the number of interconnected elements. Meshes fare better since the network diameter is proportional to the perimeter of the mesh and scales in the square root of the node count. However, a large fraction of the latency and energy in a mesh is due to the router at each hop, thus motivating the need for a more scalable topology.

One solution proposed by researchers is concentration, which reduces the total number of network nodes by sharing each network interface among multiple terminals via a crossbar switch [1]. A mesh network employing 4-way concentration leads to a 4x reduction in the effective node count (Figure 1(a)) at the cost of higher router complexity. Compared to the original network, a concentrated mesh has a smaller diameter and, potentially, a diminished area footprint. While concentration is an enabling element in the design of scalable networks, it is not sufficient by itself due to poor scalability of crossbar interconnects in its routers.

A recently proposed topology called the flattened butterfly maps a richly connected butterfly network onto a two-dimensional substrate using a two-level hierarchical organization [10]. In the 64 node network, shown in Figure 1(b), the first level employs 4-way concentration to connect the processing elements, while the second level uses dedicated links to fully connect each of the four concentrated nodes in each dimension.

The flattened butterfly is a significant improvement over the concentrated mesh in that it reduces the maximum number of hops to two, minimizing the overall impact of router delay, despite a small increase in router latency. It also makes better use of the on-chip wire bandwidth by spreading it over multiple physical channels. Unfortunately, the flattened butterfly itself is not truly scalable, as the physical channel count in each dimension grows quadratically with the number of nodes in the dimension. In addition, the use of a large number of dedicated point-to-point links and the resulting high degree of wire partitioning leads to low channel utilization, even at high injection rates. Although channel utilization can be improved through the use of non-minimal paths, this approach requires a more complex routing and buffer reservation scheme, and increases network power consumption [10].

3 Multidrop Express Channels

Planar silicon technologies are best matched to two-dimensional networks augmented with express channels for

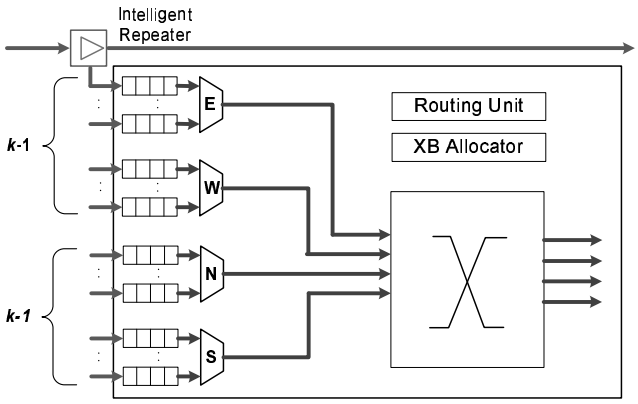


Figure 2. MECS router microarchitecture.

improved connectivity and latency reduction. While minimizing the hop count is important, as intermediate routers are the source of significant delay and energy overhead, increasing connectivity through the addition of point-to-point links leads to unscalable channel count, high serialization latencies, and low channel utilization. To address these constraints, we introduce Multidrop Express Channels – a one-to-many communication fabric that enables a high degree of connectivity in a bandwidth-efficient manner.

3.1 Overview

Multidrop Express Channels (MECS) are based on point-to-multipoint unidirectional links that connect a given source node with multiple destinations in a given row or column. The high degree of connectivity afforded by each MECS channel enables richly connected topologies with fewer bisection channels and higher per-channel bandwidth than point-to-point networks with an equivalent set of connections. Figure 1(c) shows a 64-node MECS network with 4-way concentration that reduces the number of bisection channels of the comparable flattened butterfly by a factor of two. The key characteristics of MECS are as follows:

- Bisection channel count per each row/column is equal to the network radix, k.
- Network diameter (maximum hop count) is two.
- The number of nodes accessible through each MECS channel is a function of the source node's location in a row/column and ranges from 1 to $k-1$.
- A node has 1 output port per direction, same as a mesh.
- The input port count is $2(k-1)$, equal to that of the flattened butterfly.

The high degree of connectivity provided by each channel, combined with the low channel count, maximizes per-channel bandwidth and wire utilization, while minimizing the serialization delay. The low diameter naturally leads to low network latencies. The direct correspondence between channel count and node count in each dimension allows MECS to be scaled to a large number of nodes, provided that the per-channel bandwidth is maintained.

3.2 Base Microarchitecture

Figure 2 depicts a base microarchitecture of a MECS router with $2(k-1)$ network inputs and four network outputs. As shown, each input port p_{in} has only one virtual channel FIFO, v. All inputs from a given direction share a single crossbar port for a total of four network interfaces and, in a 4-way concentrated topology, four local switch interfaces. This organization keeps the crossbar complexity low, minimizing area and delay. Arbitration complexity is comparable to a conventional mesh router with an equivalent $p_{in} \cdot v$ product. While the router design can have many variations, such as adding virtual channels or extra crossbar ports, this paper evaluates MECS networks with simple routers.

The MECS design can conserve channel power by not propagating flits beyond their destination. We augment the repeaters closest to each node with simple address decode logic to determine whether the flit should be repeated. The decoder examines the node address of the head flit and either passes all of the message's flits down the channel or routes them to the local node, inhibiting further propagation.

3.3 Analysis

Table 1 compares characteristics of the concentrated mesh (CMesh), flattened butterfly, and MECS topologies using several metrics. For each topology, the first column (in gray) provides analytical expressions for computing parameter values. The second column for each topology quantifies the parameters for a 4-ary mesh with 4-way concentration (64 terminals), and the third column repeats the analysis for a 8-ary mesh also with 4-way concentration (256 terminals). A few trends are worth highlighting:

Network diameter: Maximum hop count in a concentrated mesh grows in proportion to network perimeter, while remaining constant in both MECS and the flattened butterfly.

Bandwidth: As the network radix doubles from four in a 64-terminal network to eight in a 256-terminal configuration, the number of bisection MECS channels in each row/column doubles from 4 to 8, while in the flattened butterfly it quadruples from 8 to 32. Doubling the row/column bandwidth for the larger network keeps constant the channel width in MECS but halves it in the flattened butterfly.

Crossbar: We approximate crossbar size as $(switch\ ports \cdot \frac{BW}{port})^2$. This value is highest for CMesh and lowest for the flattened butterfly. While crossbars in the flattened butterfly have significantly more ports than those in other topologies, their area is small because crossbar bandwidth in the flattened butterfly is only a fraction of the bisection bandwidth. MECS topologies have considerably higher per-channel bandwidth than the flattened butterfly, but since the number of crossbar ports in MECS routers is

Table 1. Comparison of Concentrated Mesh (CMesh), Flattened Butterfly, and MECS topologies.

		CMesh			Flattened Butterfly			MECS		
Network	Network size		64	256		64	256		64	256
	Network radix, k		4	8		4	8		4	8
	Concentration, c		4	4		4	4		4	4
	Network diameter	$2(k-1)$	6	14	2	2	2	2	2	2
BW	Bisection BW, B_B		4,608	18,432		4,608	18,432		4,608	18,432
	Row/col channels	2	2	2	$k^2/2$	8	32	k	4	8
	BW/channel, w		576	1152		144	72		288	288
Router	Input ports, p_{in}	4	4	4	$2(k-1)$	6	14	$2(k-1)$	6	14
	Output ports, p_{out}	4	4	4	$2(k-1)$	6	14	4	4	4
	Crossbar complexity $((p_{out}+c)\cdot w)^2$		21.2e6	84.9e6		2.1e6	1.7e6		5.3e6	5.3e6
	VCs per p_{in}, α		8	8		1	1		1	1
	VC depth, β		5	5		10	15		10	15
	Buffer size, *bits* $p_{in}\cdot w\cdot\alpha\cdot\beta$		92,160	184,320		8,640	15,120		17,280	60,480
Perf, Energy	Avg hops/packet, H (random traffic)		2.54	5.25		1.52	1.76		1.52	1.76
	Avg latency/pkt, *cycles*		12.2	23.2		10.9	16.6		9.8	13.1
	Avg energy/pkt, *nJ* $E_{links}+E_{routers}$		0.83	1.57		0.44	0.76		0.52	0.93

low, the MECS crossbar area is only modestly higher than that of the flattened butterfly and significantly lower than that of CMesh. Both MECS and the flattened butterfly are amenable to crossbar optimizations that can further reduce complexity by eliminating unnecessary switch ports from the routers.

Buffering: To estimate the buffer requirements, we assume that the CMesh requires a relatively high number of VCs to avoid head-of-line blocking [1]. Both the flattened butterfly and MECS topologies can tolerate one VC per port, mitigating the adverse effects of head-of-line blocking through multiple ports. This organization also keeps arbitration complexity manageable in these high-radix routers. The depth of each VC is set to cover the maximum round-trip credit return latency. Thus, both the flattened butterfly and MECS require greater buffer depth than the CMesh to cover the wire delays associated with longer channels. While several router location-specific VC depth optimizations are possible, we assume uniform VC depths in this study.

With these assumptions, the CMesh requires by far the most buffer space, followed by MECS and the flattened butterfly. The flattened butterfly has relatively low buffer requirements because only a fraction of the bisection bandwidth reaches each router due to the high degree of channel partitioning. As the network is scaled to a larger number of nodes, the per-channel bandwidth shrinks as the port count grows, leading to a slower growth in buffer requirements. In contrast, the amount of per-channel bandwidth stays flat in MECS; as the channel count reaching each router grows, so do the buffer requirements.

Energy and Latency: To estimate the energy requirements and performance potential of each topology, we assume a uniform random packet distribution and employ energy models for wires and routers described in Section 5. The bottom rows of Table 1 show the expected latency and energy of a single packet in an unloaded network, based on the average number of hops in each topology.

With 64 nodes, the CMesh experiences a higher transmission latency than either high-radix topology due to its higher hop count. MECS, on the other other hand, observes the lowest latency, as it enjoys the same low hop count of the flattened butterfly and a decreased serialization cost due to wider channels. Scaled to 256 nodes, the latency for CMesh nearly doubles with the hop count, while latencies for the flattened butterfly and MECS increase by 52% and 34%, respectively. The gap in per-packet latency between MECS and flattened butterfly widens to 3.4 cycles as a result of increased serialization in the flattened butterfly topology, giving MECS a 20% latency advantage.

Energy results track our analytical estimates of complexity and show that the CMesh is the least efficient topology, consuming nearly 61% more energy than MECS and 88% more than the flattened butterfly in the 64-terminal network. The latter is the most energy-frugal topology of the three, a

direct result of small hop count and low crossbar complexity.

3.4 Multicast and Broadcast

Parallel computing systems often provide hardware support for collective operations such as broadcast and multicast [18, 6]. MECS can easily be augmented to support these collective operations with little additional cost because of the multipoint connectivity. A full broadcast can be implemented in two network hops by first delivering the payload to all of the network nodes in a single dimension connected to the MECS channel. Each of those nodes then broadcasts the payload to all of its siblings in the second dimension. Further discussion of these features are beyond the scope of this paper; here we focus on traditional NOC workloads that require point-to-point communication.

3.5 Related Work

While MECS bear some resemblance to conventional multi-drop (broadcast) buses, a bus is an all-to-all medium, whereas MECS is a one-to-many topology. The YARC router [17] employed an 8x8 grid of switches to implement a radix-64 router. Each switch was connected to other switches in a given row via a dedicated bus, making it a one-to-many configuration similar to MECS. In each column, point-to-point channels connected each switch to others, analogous to the flattened butterfly configuration. The difference between YARC and MECS is our use of uni-directional one-to-many channels in both dimensions with intelligent repeaters for conserving power. This gives MECS desirable scalability properties in terms of performance and energy efficiency.

Kim et al. [10] proposed to extend the flattened butterfly with non-minimal routing through the use of bypass links, which provide additional exit points in an otherwise point-to-point channel. This bypassing is similar in nature to the multi-drop capability in MECS. However, its use in the flattened butterfly network requires a complex reservation protocol as input ports are shared between regular and bypassed channels. MECS do not need special routing, have dedicated input ports, and require significantly fewer channels.

Finally, Express Virtual Channels [11] attempt to reduce the latency and energy overhead of routers in a mesh topology through an aggressive flow control mechanism. MECS is a topology which also aims to eliminate the impact of intermediate routers and has a broader objective of making efficient use of the available on-chip wire budget.

4 Generalized Express Cubes

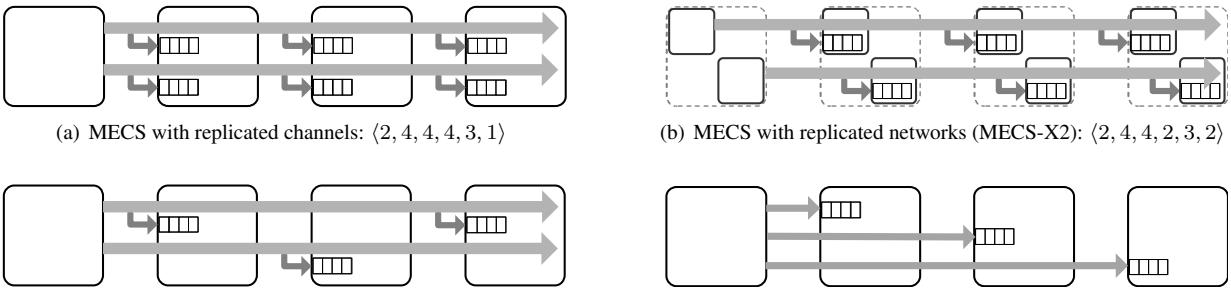
Due to constraints imposed by planar silicon, scalable NOC topologies are best mapped to low-dimensional k-ary n-cubes augmented with express channels and concentration. Other NOC topologies map well to this basic organization; for instance, the flattened butterfly can be viewed as a concentrated mesh with express links connecting every node with all non-neighboring routers along the two dimensions. This section explores the resulting space of topologies, which we refer to as Generalized Express Cubes (GEC), and includes both MECS and the flattened butterfly.

Building on the k-ary n-cube model of connectivity, we define the six-tuple $\langle n, k, c, o, d, x \rangle$ as:
n - network dimensionality
k - network radix (nodes/dimension)
c - concentration factor (1 = none)
o - router radix (output channels/dimension in each node)
d - channel radix (sinks per channel)
x - number of networks

The router radix, o, specifies the number of output channels per network dimension per (concentrated) node, equalling two in a mesh (one in each direction) and three in the flattened butterfly of Figure 1(b). The channel radix, d, specifies the maximum number of sink nodes per channel. A value of one corresponds to point-to-point networks; greater values define MECS channels. Finally, replicated topologies, which allow bandwidth to be distributed among several networks, can be expressed via the x parameter. Using this taxonomy, the six-tuple for the 64-terminal MECS network from Figure 1(c) is $\langle 2, 4, 4, 2, 3, 1 \rangle$, indicating that the baseline topology is a single-network 4-ary 2-cube employing 4-way concentration, with radix-4 routers and up to three nodes accessible via each channel. In general, we note that this taxonomy is not sufficient to specify the exact connectivity for a large set of networks encompassed by the GEC model. Here, our intent is to focus on a small set of regular topologies attractive for on-chip implementation with sufficiently diverse characteristics.

Figure 3 shows several possible topologies (one-dimensional slices) that can be specified with the GEC model, with MECS (from Figure 1(c)) at one end of the spectrum and the flattened butterfly (Figure 3(d)) at the other. In wire-rich NOCs, channel bandwidth may be wasted when link width exceeds the size of a frequently-occurring but short packet type, such as a read request or a coherence transaction. Wide channel topologies, like CMesh and MECS, are vulnerable to this effect while narrower channel topologies, such as the flattened butterfly, are less susceptible. Partitioning the bandwidth across multiple channels can reduce this type of wasted bandwith.

One means of partitioning, shown in the $\langle 2, 4, 4, 4, 3, 1 \rangle$ network of Figure 3(a), divides each baseline MECS channel into two, each with one-half of the bandwidth. This configuration can improve bandwidth utilization and reduce head-of-line blocking for short packets by doubling the router radix, o. Latency for long packets, however, might suffer as a result of increased serialization delay. Another potential downside is an increase in arbitration complexity due to the doubling in the number of ports.



(a) MECS with replicated channels: $\langle 2, 4, 4, 4, 3, 1 \rangle$

(b) MECS with replicated networks (MECS-X2): $\langle 2, 4, 4, 2, 3, 2 \rangle$

(c) Partitioned MECS (MECS-P2): $\langle 2, 4, 4, 4, 2, 1 \rangle$

(d) Fully-partitioned MECS as flattened butterfly: $\langle 2, 4, 4, 3, 1, 1 \rangle$

Figure 3. MECS variants for cost-performance trade-off. Only one dimension is shown for simplicity.

An alternative approach replicates the networks, increasing x, such that each network has full connectivity of the original but with a fraction of the bandwidth. The resuling $\langle 2, 4, 4, 2, 3, 2 \rangle$ topology is shown in Figure 3(b). An advantage of such a design is that it does not increase the router radix and reduces the combined crossbar area. While replicated topologies have been proposed in the past to exploit greater bandwidth with a given router design [1, 9], our work shows that replication in wire-rich substrates can yield significant throughput gains and energy savings for a given bisection bandwidth.

A third option that enables a more aggressive area reduction in a MECS topology at a cost of reduced performance partitions each multidrop channel into two (or more), interleaving the destination nodes among the resulting links. The $\langle 2, 4, 4, 4, 2, 1 \rangle$ network of Figure 3(c) increases the router radix o and decreases the channel radix d. This *partitioned* MECS, or MECS-P2, topology has reduced network buffer requirements proportional to the partitioning factor and can decrease router crossbar complexity.

In the limit, completely partitioning a MECS topology yields a point-to-point network, such as the $\langle 2, 4, 4, 3, 1, 1 \rangle$ flattened butterfly in Figure 3(d). While further analysis of networks in the space of Generalized Express Cubes is beyond the scope of this paper, the experimental results in Section 6 include three of the networks of Figure 3.

5 Experimental Methodology

Topologies: To compare the different topologies, we used a cycle-precise network simulator that models all router pipeline delays and wire latencies. We evaluated the mesh, concentrated mesh, flattened butterfly, and MECS topologies. We also considered two topologies with replicated networks, CMesh-X2 and MECS-X2, and a partitioned MECS variant called MECS-P2. For the largest simulated network size, we considered a variant of the flattened butterfly that limits the maximum channel span to four nodes. This topology, called FBfly4, reduces the number of bisection channels in exchange for increased the per-channel bandwidth, thus trading serialization latency for a greater network diameter.

Network parameters: We considered network sizes of 64 and 256 terminals. Except for the mesh, all topologies use 4-way concentration, reducing the effective node count to 16 and 64, respectively. Where applicable, parameters for various topologies and network sizes are the same as those in the analytical comparison of Section 3.3. Table 2 summarizes the simulated configurations.

As the bisection bandwidth across all topologies is kept constant, the concentrated mesh has twice the per-channel bandwidth as the basic mesh, while the flattened butterfly, MECS, and all replicated and partitioned topologies evenly distribute this bandwidth among their links. All of the networks employ dimension-order routing (DOR), resulting in minimal length and hop-count paths.

We assume a router latency of 2 cycles in a mesh and 3 cycles in CMesh, FBfly and MECS topologies regardless of network size. All topologies employ look-ahead routing. Mesh and CMesh routers use speculative switch allocation, overlapping VC and switch allocation in the first stage of the pipeline. As both FBfly and MECS routers have just one VC per physical port, they are essentially wormhole and do not require VC allocation. Switch allocation for an 18-port wormhole router required by these topologies in the 256-terminal configuration fits comfortably in a single 20 FO4 cycle [14]. All topologies except the mesh require an extra cycle for switch setup in the second stage of the pipeline. The final pipeline stage for all configurations is switch traversal.

Synthetic workloads: Our synthetic workloads consist of three traffic patterns: bit complement, uniform random, and transpose – permutations that exhibit diverse behaviors. The packet sizes are stochastically chosen as either short 64-bit packets, typical of requests and coherence transactions, or long 576-bit packets, representative of replies and writes.

Application evaluation: To simulate full applications, we use traces from parallel application benchmark suites. We first examined traces from the Splash-2 suite [26, 21] obtained from a shared-memory CMP system simulator [11]. We also generated traces from a subset of the PARSEC parallel benchmark suite [2] using the M5 simulator. In M5, we model a 64-core CMP with the Alpha ISA and a modified Linux OS [3]. Table 3 summarizes our system configuration, comprised of two-way out-of-order cores with private L1 instruction and data caches, a shared NUCA L2 cache

Table 2. Simulated network configurations.

	64 nodes	256 nodes
Traffic patterns	bit complement, uniform random, transpose	
Traffic type	64- and 576-bit packets, stochastic generation	
Topology rows x cols x concentration	8x8x1: Mesh 4x4x4: CMesh, CMesh-X2 FBfly MECS, MECS-X2	16x16x1: Mesh 8x8x4: CMesh, CMesh-X2 FBfly, FBfly4 MECS, MECS-X2, MECS-P2
Channel BW (bits)	Mesh: 288 FBfly: 144 CMesh: 576 MECS: 288 CMesh-X2: 288 MECS-X2: 144	Mesh: 576 FBfly: 72 CMesh: 1152 MECS: 288 CMesh-X2: 576 MECS-X2: 144 FBfly4: 115 MECS-P2: 144
Router latency (cycles)	Mesh: 2; CMesh*, FBfly, MECS*: 3	Mesh: 2; CMesh*, FBfly*, MECS*: 3
VCs/channel	Mesh, CMesh*: 8; FBfly, MECS*: 1	Mesh, CMesh*: 8; FBfly* MECS*: 1
Buffers/VC	Mesh, CMesh*: 5; FBfly, MECS*: 10	Mesh, CMesh*: 5; FBfly*, MECS*: 15

Table 3. Full-system configuration.

Cores	64 on-chip, Alpha ISA, 2GHz clock, 2-way out-of-order, 2 integer ALUs, 1 integer mult/div, 1 FP ALU, 1 FL mult/div
L1 cache	32KB instruction/32KB data, 4-way associative, 64B lines, 3 cycle access time
L2 cache	fully shared S-NUCA, 16MB, 64B lines, 8-way associative, 8 cycle/bank access time
Memory	150 cycle access time, 8 on-chip memory controllers
PARSEC applications	Blackscholes, Bodytrack, Canneal, Ferret, Fluidanimate, Freqmine, Vip, x264

Table 4. Router energy per 576-bit packet (pJ).

64-terminal network	Mesh	CMesh	CMesh X2	FBfly	MECS	MECS X2
Buffers	61.7	61.6	61.7	36.0	35.9	36.0
Crossbar	78.0	228.8	120.7	81.6	135.0	74.2
Arbiters	1.2	1.1	1.8	2.4	1.5	2.5

and eight on-die memory controllers. All benchmarks in Table 3 were run with sim-medium input sets with the exception of *blackscholes*, which was simulated with sim-large. The remaining PARSEC benchmarks are currently incompatible with our simulator.

We capture all memory traffic past the L1 caches for replay in the network simulator. While we correctly model most of the coherence protocol traffic, certain messages associated with barrier synchronization activity are absent from our traces, as their network-level behavior artificially interferes with traffic measurements in the simulator. These messages constitute a negligible fraction of network traffic.

Energy: We use a combination of CACTI 6.0 [13] and Orion [24] to derive energy consumption in the channels and routers for 64- and 256-terminal networks in a 45 nm technology. For MECS topologies, we conservatively model an asymmetric crossbar that directly connects all inputs to outputs. We expect this configuration to be less energy-efficient than the smaller symmetric crossbar and input mux combination in Figure 2. Channels employ energy-optimized repeated wires with a 30% delay penalty, consuming 97 fJ per millimeter of wire. Table 4 summarizes the energy expended by a single 576-bit message in each router component for a 64-node network.

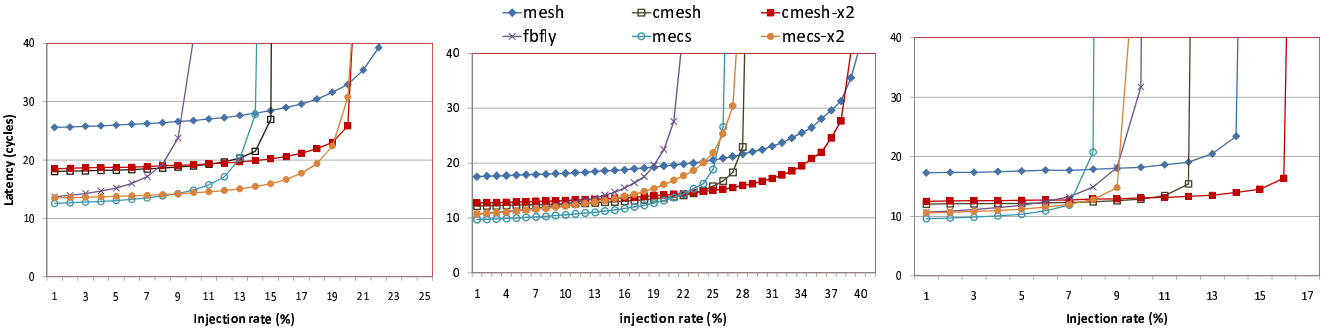
6 Evaluation

6.1 Synthetic Workload - 64 nodes

Figure 4 summarizes the evaluation of a 64 node system with the three synthetic traffic patterns. In general, we observe that the mesh has the highest latency at low loads, exceeding that of other topologies by 40-100%. The concentrated mesh has the second-highest latency, trailing the flattened butterfly by 14-34%. The baseline MECS topology consistently has the lowest latency at low injection rates, outperforming FBfly by 9%, on average. MECS-X2 has zero-load latencies comparable to those of the flattened butterfly.

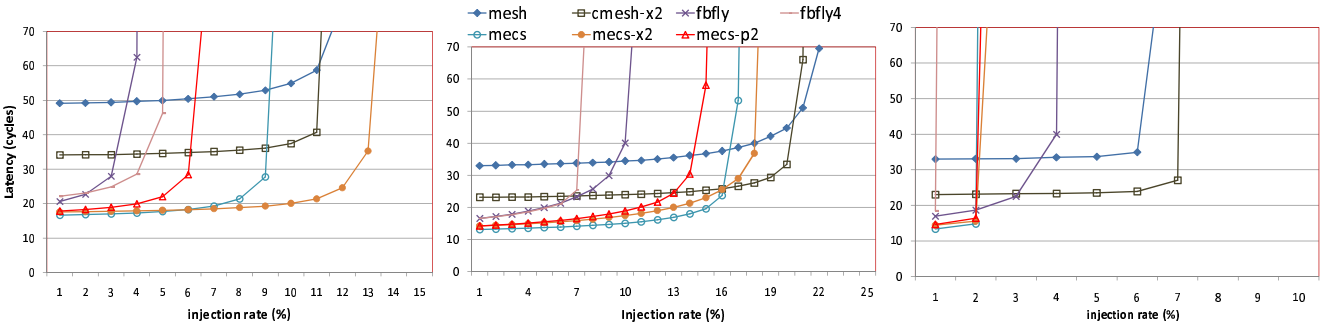
The results are consistent with our expectations. The mesh has a high hop count, paying a heavy price in end-to-end router delay. The CMesh improves on that by halving the network diameter, easily amortizing the increased router latency. The flattened butterfly and MECS-X2 have the same degree of connectivity, same number of bisection channels, and same bandwidth per channel; as such, the two topologies have similar nominal latencies. Finally, the single-channel MECS has the same connectivity as the flattened butterfly but with twice as much per-channel bandwidth, which results in the lowest zero-load latency.

The picture shifts when one considers the throughput of different topologies. The mesh, due to its high degree of pipelining, yields consistently good throughput on all three workloads. CMesh-X2 restores the channel count lost in the baseline CMesh due to concentration, effectively matching the throughput of the basic mesh as a result. The flattened butterfly has the lowest throughput on two of the three traffic patterns as it cannot effectively utilize all of the available channels with dimension-order routing. MECS and MECS-



(a) Bit Complement Traffic (b) Uniform Random Traffic (c) Transpose Traffic

Figure 4. Load-latency graphs for 64-node mesh, CMesh, flattened butterfly and MECS topologies.



(a) Bit Complement Traffic (b) Uniform Random Traffic (c) Transpose Traffic

Figure 5. Load-latency graphs for 256-node mesh, CMesh, flattened butterfly and MECS topologies.

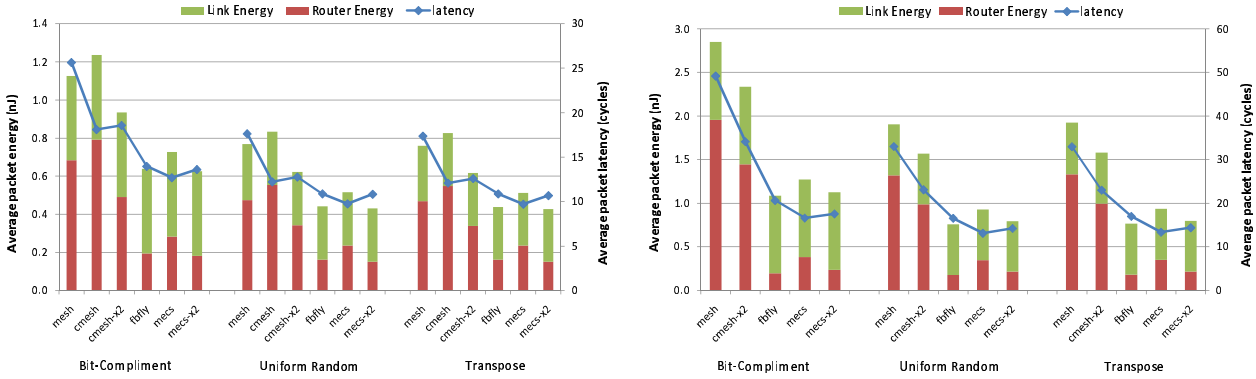
X2 fall in the middle, although the latter enjoys higher throughput than the basic MECS on all three permutations.

The transpose traffic pattern deserves a separate look, as it represents an adversarial scenario for many topologies under dimension-order routing. This permutation mimics a matrix transpose operation, in which all nodes from a given row send messages to the same column. In MECS, packets from different source nodes in each row arrive at the "corner" router via separate channels but then serialize on the shared outbound link, compromising throughput. The turn node is a bottleneck in mesh and CMesh topologies as well; however, the mesh benefits from the lack of concentration, so the traffic is spread among more channels, while the CMesh enjoys very wide channels that help throughput. Finally, the flattened butterfly achieves better throughput than MECS by virtue of its high-radix switch, which effectively provides a dedicated port for each source-destination pair. In all cases, throughput can be improved through the use of improved routing policies, which we discuss in Section 7.1.

6.2 Synthetic Workload - 256 nodes

In the larger network, the basic mesh becomes decidedly unappealing at all but the highest injection rates due to enormous zero-load latencies. CMesh-X2 also sees its latency rise significantly, exceeding that of the flattened butterfly and MECS by 35-105% at low injection rates. In both the mesh and CMesh-X2, the degradation is due to the large increase in the average hop count. We do not present the results for the basic CMesh topology, as it is vastly inferior to CMesh-X2 in terms of throughput and energy in a network this size. As expected, all MECS variants enjoy the lowest latency at low loads due to a good balance of connectivity, channel count, and channel bandwidth. As such, they outperform the flattened butterfly by 14-20% in terms of latency. Interestingly, FBfly4 has a slightly lower zero-load latency than the basic flattened butterfly, which means that the serialization component of the latency in the flattened butterfly dominates the increased hop count in FBfly4.

In terms of throughput, CMesh-X2 and the mesh show the highest degree of scalability. MECS-X2 is also very competitive on two of the three patterns, with transpose being the same exception as earlier. In fact, on the bit-complement permutation, MECS-X2 has the highest throughput of any topology considered here. The partitioned MECS, MECS-P2, generally performs worse than the other MECS variants. However, it nearly always outperforms the flattened butterfly and FBfly4 in terms of both latency and throughput. The sole exception is transpose, on which the flattened butterfly achieves higher throughput. Thus, MECS-P2 appears attractive for large networks that are sensitive to area, energy and latency but have modest bandwidth requirements. Finally, we observe that both flattened butterfly topologies tend to saturate quite early, as they are unable to keep all of the channels utilized, thereby wasting network bandwidth.



(a) 64-terminal network

(b) 256-terminal network

Figure 6. Per-packet energy and latency on synthetic traffic.

6.3 Synthetic Workload Energy

To evaluate the relative energy efficiency of the different topologies, we simulated 100,000 packets for each configuration at a nominal 1% injection rate, collecting latency and energy data. Figure 6 summarizes the results. In the 64-terminal network, the CMesh consumes the most energy due to the relatively large crossbar at each network hop. The mesh, despite its higher hop count, is more energy efficient as a result of its compact, low-radix routers. CMesh-X2 combines low hop count with a crossbar that is smaller than that in the single-network CMesh to yield an even lower energy footprint and a better energy-delay product. MECS, while not the most energy-efficient topology, is quite competitive by virtue of having crossbar complexity comparable to that of the CMesh-X2 router, but requiring fewer network hops per packet. MECS delivers nearly 30% savings in router energy and 14% in total network energy over CMesh-X2. The most energy-frugal topologies are the flattened butterfly and MECS-X2, which combine low hop count and small crossbars to yield the lowest network energy.

Similar trends are observed in the larger, 256-terminal, network. The gap between mesh and express channel topologies grows as a larger fraction of the energy in mesh and CMesh networks is expended in routers due to increased network diameter. In fact, the energy-efficiency of express channel topologies starts to approach that of an ideal network [11] with just 18-38% of total network energy dissipated in routers and the rest in channels.

6.4 Application-based Workloads

Figure 7 shows the relative performance and energy of various topologies on our PARSEC trace-driven workloads. The results reflect total network energy and average per-packet latency. Because the injection rates in the simulated applications are low, latency appears to be a more stringent constraint than bandwidth for evaluating the topologies. MECS has the lowest latency, consistently outperforming the flattened butterfly and MECS-X2 by nearly 10%. The mesh has by far the highest latency as a result of its high hop count, followed by the CMesh-X2 and the basic CMesh. Because the results on the Splash-2 workloads mirror those for PARSEC, we omit them from this manuscript.

Energy trends also track closely the results of the synthetic benchmarks. The mesh, while more energy-efficient than CMesh, has a higher energy-delay product (not shown in the graph) as a result of its high network latency. CMesh-X2 has an energy-delay product that is 23% lower, on average, than that of the basic concentrated mesh due to its significantly lower crossbar energy. The flattened butterfly and MECS-X2 are the most energy-efficient topologies and enjoy the lowest energy-delay product. MECS, the topology with the lowest latency, has an energy-delay product that is within 10% of that of MECS-X2, the topology with the lowest energy-delay. The results confirm that MECS successfully minimizes latency, while replication and partitioning are effective at minimizing network energy.

7 Discussion

7.1 Adaptive Routing

Adaptive routing can improve network load balance and boost throughput by smoothing out traffic non-uniformities [20]. To evaluate its impact across the topologies, we focused on a family of adaptive routing algorithms based on O1Turn [19]. We consider both the original (statistical) approach and a number of adaptive variants that use various heuristics to estimate network congestion and choose the first dimension in which to route. Among the heuristics considered were the degree of link multiplexing, downstream VC and credit availability, and a simple variant of RCA [7]. After evaluating each routing policy on every topology, we picked the algorithm that performed best for a given topology across all three of our synthetic traffic patterns.

The topologies see little benefit from adaptive routing on uniform random and bit-complement traffic patterns. However, Figure 8 shows that all topologies demonstrate a substantial throughput improvement under the transpose permutation. As the deadlock avoidance strategy requires two VCs

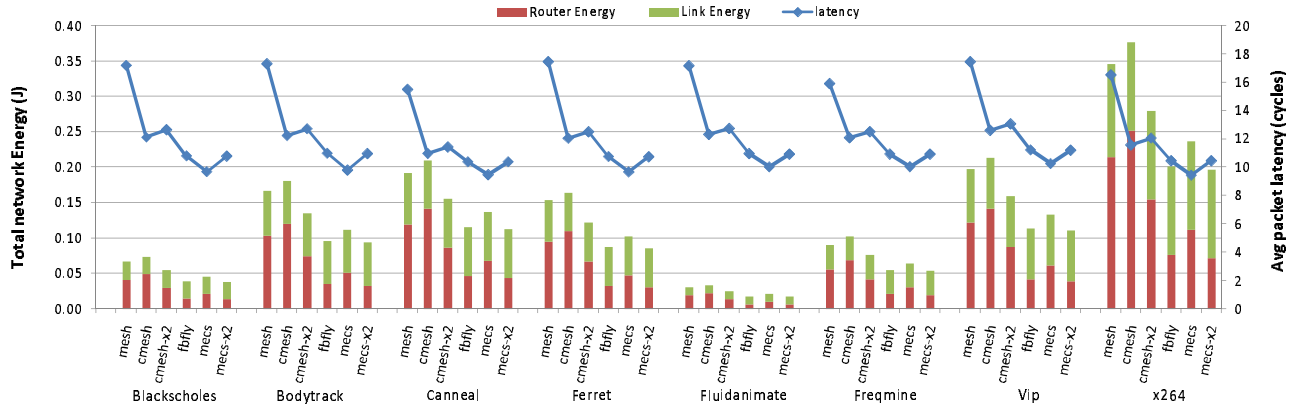


Figure 7. Performance and energy efficiency of different topologies on the PARSEC suite.

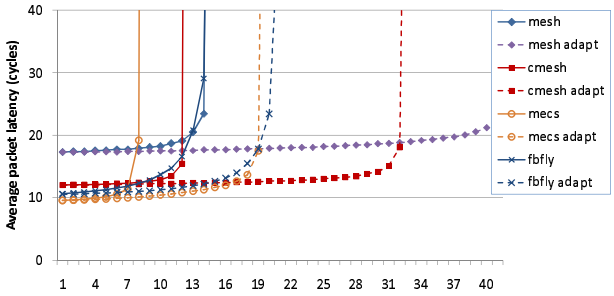


Figure 8. Routing policies on *transpose*.

per channel with adaptive routing, we augment the baseline DOR-routed flattened butterfly and MECS topologies with a second VC to isolate the performance improvement that stems from the routing algorithm.

The mesh achieves the highest throughput relative to other topologies, as the lack of concentration allows each source to use a YX route without interference from any other node in the network. The CMesh, despite having wider channels, does not have this luxury, as all terminals of a single concentrated node share the same set of output ports. The flattened butterfly and MECS have the same limitation as CMesh, but also have narrower channels, thereby saturating at a lower injection rate. MECS is able to cover most of the gap relative to the flattened butterfly, almost matching its throughput.

Because the best routing algorithm is typically tied closely to the characteristics of the topology, comparing all of the topologies using O1Turn derivatives is neither complete nor completely fair. For example, the path diversity available in the flattened butterfly motivated the authors of that paper to use a non-minimal adaptive routing algorithm in their evaluation [10]. Because non-minimal routing incurs design complexity overheads and energy penalties for non-minimally routed packets, adaptive routing algorithms must balance throughput needs and energy targets of NOCs. Results on the PARSEC benchmarks suggest that these real-world applications have modest injection rates, implying that NOCs may be more sensitive to latency and energy than throughput.

7.2 Scaling beyond 256 terminals

As Section 6 demonstrates, express channel topologies scale better to 256 terminals in terms of performance and energy-efficiency than plain k-ary 2-cubes. In turn, Table 1 reveals that area overhead of express channel on-chip interconnects is also quite competitive. However, even under the assumption that growth in available wire bandwidth can keep up with Moore's law, it is unlikely that any of these topologies can be scaled unmodified up to 1000 terminals or beyond.

The flattened butterfly suffers from an explosion in channel count which impacts serialization latency, throughput and arbitration complexity at larger network sizes. In MECS, an important impediment to scalability is the asymmetry between input and output bandwidth at each router node. As the network size is scaled up, the output bandwidth at each MECS router represents an ever-shrinking fraction of the input bandwidth, causing increased contention for output ports, higher packet latencies, and reduced throughput.

Kim et al. suggested several approaches to scaling the flattened butterfly, which could be appropriate for other NOC interconnects, including MECS. These include increasing the degree of concentration, increasing the dimensionality of the network, and a hybrid approach to scaling [10]. While an evaluation of these techniques is beyond the scope of this paper, further innovation may be required to compose topologies that can be comfortably scaled to thousands of nodes within the constraints imposed by planar silicon technology and electrical interconnects.

8 Conclusion

Designing a scalable NOC fabric requires balancing performance, energy consumption, and area. To address these constraints, this paper introduced a new family of networks called Multidrop Express Channels (MECS) which are composed of point-to-multipoint unidirectional links. The resulting network enjoys a high-degree of inter-node connectivity, low hop count, and bisection channel count that is propor-

tional to the arity of the network dimension. Compared to the other topologies we evaluate, MECS provides superior low-load latency and competitive energy efficiency.

In the broader perspective, we observe that MECS belongs to a larger class of networks expressible via Generalized Express Cubes – a framework that extends k-ary n-cubes with concentration and express channels. We explore several GEC-expressible topologies, including the flattened butterfly, establishing area, energy and performance advantages of various configurations that differ in channel count, connectivity and bandwidth. We expect that further research in this space of networks will provide additional insight into and solutions for scalable NOCs.

Acknowledgments

This research is supported by NSF CISE Infrastructure grant EIA-0303609 and NSF grant CCF-0811056. Part of this work was performed when Dr. Mutlu was a researcher and Mr. Grot a research intern at Microsoft Research.

References

[1] J. D. Balfour and W. J. Dally. Design Tradeoffs for Tiled CMP On-chip Networks. In *International Conference on Supercomputing*, pages 187–198, June 2006.

[2] C. Bienia, S. Kumar, J. P. Singh, and K. Li. The PARSEC Benchmark Suite: Characterization and Architectural Implications. Technical Report TR-811-08, Princeton University, January 2008.

[3] N. L. Binkert, R. G. Dreslinski, L. R. Hsu, K. T. Lim, A. G. Saidi, and S. K. Reinhardt. The M5 Simulator: Modeling Networked Systems. In *IEEE Micro*, pages 52–60, July/August 2006.

[4] W. J. Dally. Express Cubes: Improving the Performance of k-ary n-cube Interconnection Networks. *IEEE Transactions on Computers*, 40(9):1016–1023, September 1991.

[5] W. J. Dally and B. Towles. *Principles and Practices of Interconnection Networks*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.

[6] A. Gara, M. Blumrich, D. Chen, G. Chiu, P. Coteus, M. Giampapa, R. Haring, P. Heidelberger, D. Hoenicke, G. Kopcsay, T. Liebsch, M. Ohmacht, B. Steinmacher-Burow, T. Takken, and P. Vranas. Overview of the Blue Gene/L System Architecture. *IBM Journal of Research and Development*, 49(2/3):195–212, 2005.

[7] P. Gratz, B. Grot, and S. W. Keckler. Regional Congestion Awareness for Load Balance in Networks-on-Chip. In *International Symposium on High-Performance Computer Architecture*, pages 203–214, February 2008.

[8] P. Gratz, C. Kim, R. McDonald, S. W. Keckler, and D. Burger. Implementation and Evaluation of On-chip Network Architectures. In *International Conference on Computer Design*, pages 477–484, October 2006.

[9] P. Gratz, K. Sankaralingam, H. Hanson, P. Shivakumar, R. McDonald, S. W. Keckler, and D. Burger. Implementation and Evaluation of a Dynamically Routed Processor Operand Network. In *International Symposium on Networks-on-Chip*, pages 7–17, May 2007.

[10] J. Kim, J. Balfour, and W. Dally. Flattened Butterfly Topology for On-chip Networks. In *International Symposium on Microarchitecture*, pages 172–182, December 2007.

[11] A. Kumar, L.-S. Peh, P. Kundu, and N. K. Jha. Express Virtual Channels: Towards the Ideal Interconnection Fabric. In *International Symposium on Computer Architecture*, pages 150–161, May 2007.

[12] R. Mullins, A. West, and S. Moore. Low-Latency Virtual-Channel Routers for On-Chip Networks. In *International Symposium on Computer Architecture*, pages 188–197, June 2004.

[13] N. Muralimanohar, R. Balasubramonian, and N. Jouppi. Optimizing NUCA Organizations and Wiring Alternatives for Large Caches with CACTI 6.0. *International Symposium on Microarchitecture*, pages 3–14, December 2007.

[14] L.-S. Peh. *Flow Control and Micro-Architectural Mechanisms for Extending the Performance of Interconnection Networks*. PhD thesis, Stanford University, 2001.

[15] L.-S. Peh and W. J. Dally. A Delay Model and Speculative Architecture for Pipelined Routers. In *International Symposium on High-Performance Computer Architecture*, pages 255–266, January 2001.

[16] D. Pham et al. Overview of the Architecture, Circuit Design, and Physical Implementation of a First-Generation Cell Processor. *IEEE Journal of Solid-State Circuits*, 41(1):179–196, January 2006.

[17] S. Scott, D. Abts, J. Kim, and W. J. Dally. The BlackWidow High-Radix Clos Network. In *International Symposium on Computer Architecture*, pages 16–28, June 2006.

[18] S. L. Scott. Synchronization and Communication in the T3E Multiprocessor. *ACM SIGPLAN Notices*, 31(9):26–36, September 1996.

[19] D. Seo, A. Ali, W.-T. Lim, N. Rafique, and M. Thottethodi. Near-Optimal Worst-Case Throughput Routing for Two-Dimensional Mesh Networks. In *International Symposium on Computer Architecture*, pages 432–443, June 2005.

[20] A. Singh, W. J. Dally, B. Towles, and A. K. Gupta. Globally Adaptive Load-Balanced Routing on Tori. *IEEE Computer Architecture Letters*, 3(1), March 2004.

[21] SPLASH-2. http://www-flash.stanford.edu/apps/SPLASH/.

[22] S. Vangal et al. An 80-Tile 1.28 TFLOPS Network-on-Chip in 65nm CMOS. In *International Solid-State Circuits Conference*, pages 98–99, February 2007.

[23] E. Waingold et al. Baring It All to Software: RAW Machines. *IEEE Computer*, 30(9):86–93, September 1997.

[24] H.-S. Wang, X. Zhu, L.-S. Peh, and S. Malik. Orion: a Power-Performance Simulator for Interconnection Networks. *International Symposium on Microarchitecture*, pages 294–305, November 2002.

[25] D. Wentzlaff et al. On-Chip Interconnection Architecture of the Tile Processor. *IEEE Micro*, 27(5):15–31, September/October 2007.

[26] S. Woo, M. Ohara, E. Torrie, J. Singh, and A. Gupta. The SPLASH-2 Programs: Characterization and Methodological Considerations. In *International Symposium on Computer Architecture*, pages 24–36, June 1995.