Appears in CMP-MSI 2008: 2nd Workshop on Chip Multiprocessor Memory Systems and Interconnects

Scalable On-Chip Interconnect Topologies

Boris Grot and Stephen W. Keckler
Department of Computer Sciences
The University of Texas at Austin
{*bgrot, skeckler*}*@cs.utexas.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. Cost and performance 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 and performance considering the advantages and limitations afforded on a die.

Our contributions are three-fold. First, we introduce a taxonomy for on-chip interconnect topologies. Second, we propose a new topology, called Multidrop Express Channels (MECS), that uses a one-to-many communication model to provide a high degree of connectivity in a bandwidth-efficient manner. And third, using a combination of analytical- and simulation-based studies, we show that MECS can provide considerable latency and throughput advantages over previously proposed topologies in a cost-competitive manner.

1 Introduction

As continuing scaling of Moore's law enables ever greater transistor densities, design complexity and power limitations of conventional out-of-order superscalars 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. Examples include the 64-core Tile Processor from Tilera [18] and Intel's 80-core Terascale chip [15], executed in 90 and 65 nm technology, respectively. With continuing technology scaling, we can expect hundreds of general and special-purpose cores integrated on a single silicon substrate in the near future.

In order to interconnect such a high number of elements on a die, researchers have turned to interconnection networks as a replacement to conventional shared buses and ad-hoc wiring solutions. On-chip interconnects are attractive due to their regularity and modular design, which can lead to better routability, electrical characteristics and fault tolerance [4]. Most existing networks on a chip (NOCs) are based on rings [13] or two-dimensional meshes [18, 15, 8, 16] – 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 into hundreds or thousands. Aggravating the situation, two-dimensional substrates restrict the space of implementable networks. In response, researchers have recently proposed *concentration* as a means to reduce the number of network nodes by co-locating multiple elements at each network interface [1]. Another solution involves *flattening* a conventional butterfly network for use on a chip [9]. 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.

Our contributions are three-fold. First, we establish a general framework for expressing the space of topologies suitable and attractive for on-chip implementation. Second, we introduce Multidrop Express Channels (MECS) – a new family of topologies based on express cubes [2] that are specifically designed to fit the unique advantages and constraints of on-chip networks. And third, we compare MECS to previously proposed topologies using an analytical model and simulation studies, establishing performance and scalability advantages provided by MECS.

Our simulation results on synthetic traffic patterns show that in a 64-node network, Multidrop Express Channels provide a latency advantage of 8-31% over previously proposed topologies with average throughput. Scaled to a 256 node configuration, MECS deliver a 16-45% latency improvement across all workloads, and throughput gain in the range

of 5% to 33% on three of the four benchmarks evaluated.

Section 2 provides a brief introduction to interconnection networks and surveys relevant prior art in on-chip interconnects. Section 3 introduces our framework for expressing the space of direct topologies in NOCs based on the notion of Generalized Express Channels. In section 4, we describe Multidrop Express Channels as a cost-effective and scalable topology for on-chip networks. We also analytically compare MECS to previously proposed topologies and propose enhancements to further improve their cost and performance. A simulation-based evaluation of MECS is presented in section 5. Finally, section 6 summarizes our contributions and outlines possible future research directions.

2 Background

In this work, we are interested in evaluating on-chip network topologies in terms of their cost, performance, and scalability. We give some consideration to energy-efficiency, but leave a more detailed analysis, along with a study of other factors such as fault-tolerance, to future work.

2.1 Cost

Traditionally, the cost of interconnection networks has been primarily dictated by pin constraints of the available packaging technology. In networks on a chip, however, die area and wiring complexity are the main determinants of network cost. The area overhead is due to routers and communication channels. Flit buffers, crossbars, and control logic are the primary contributors to the routers' area cost. However, since control logic has a negligible footprint [7, 8], we will restrict our analysis to crossbar and buffer overheads. Assuming that wires are routed in higher-level metal layers over active components, only the cost of repeaters needs to be accounted for when estimating the area of communication channels.

The wiring complexity, combined with restrictions imposed by planar silicon substrates, profoundly impacts 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 straight-forward to implement in silicon, have short channel lengths and low router complexity. On the other hand, conventional highly-dimensional k-ary n-cube topologies are usually unattractive as they are either impossible to build on 2D substrates or require some form of flattening. Flattening of conventional k-ary n-cubes can lead to non-minimal channel lengths, thus adversely impacting wire delay and energy, and can complicate the routability of the design. Traditional indirect topologies, such as fat trees and clos networks, are also unattractive for the same reasons.

2.2 Performance

The performance of interconnection networks is determined by two factors: throughput and latency [3]. 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. Two components make up packet latency; these are the header latency, T_h, and the serialization delay, T_s.

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

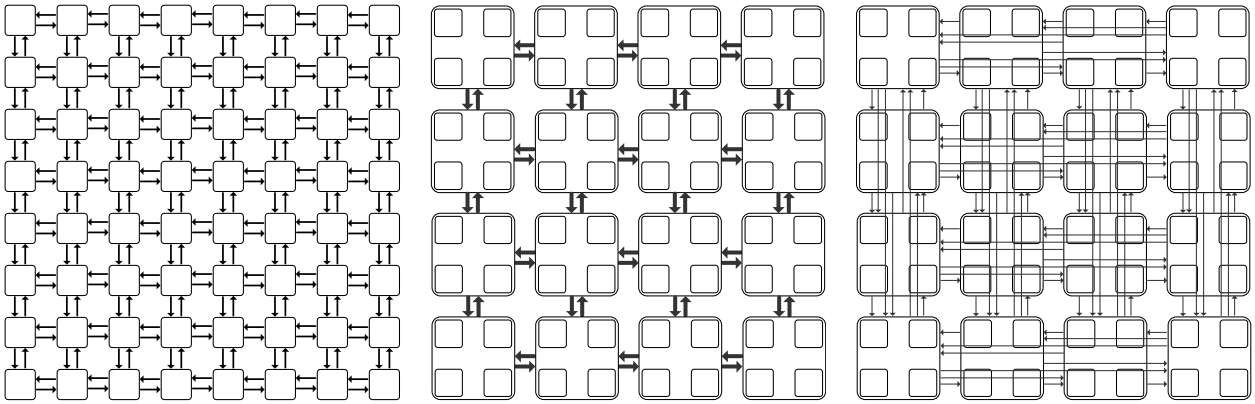
The header latency is the sum of router delay, d_r, and wire delay, d_w, at each hop, multiplied by the hop count, H. The serialization latency is the number of cycles required by a message to cross the channel and is simply the quotient of the message length, L, and the channel width, W. The resulting expression, above, 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.

By far, the most popular NOC topology to date has been a two-dimensional mesh [18, 15, 8, 16]. 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 [12, 11] and express virtual channels [10] to reduce the router latency to one or two cycles per hop. But with single-cycle channel delays, router latency in two-dimensional meshes remains a major component of network latency.

2.3 Energy

On-chip network power has been estimated to consume up to 28% of total chip power [15]. Excluding the clock tree, most of the energy expanded in NOCs is due to channels, router fifos and router crossbar fabrics. In a two-dimensional mesh, each of these is responsible for 15-30% of network power consumption [15, 17]. Thus, roughly 30% to 60% of per-hop power is dissipated in routers, contributing to a chip-wide power drain of up to 16%.

One-dimensional ring networks have simpler routers by virtue of having fewer network ports; as such, the routers can be expected to consume less energy than those in a mesh. Unfortunately, rings have a high average hop count



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

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

and any energy savings gained through low router complexity are likely offset by larger number of network hops.

2.4 Scalability

Given that today, the most aggressive CMP designs have dozens of cores, it is only a matter of time until CMPs featuring hundreds or thousands of processing elements enter the main stream. As such, it is important to consider how today's on-chip interconnect fabrics will scale into tomorrow on the basis of cost, performance and energy.

While cost-effective, simple rings appear to be the least scalable alternative, since the hop count - and thus, latency and energy - grow linearly with the number of interconnected elements.

Meshes fair better, as the network diameter is proportional to the perimeter of the mesh, and thus scales with the square root of mesh size. However, when one considers that at least half of the latency and a large fraction of the energy required in a mesh is due to the router at each hop, the need for a more scalable topology becomes apparent.

One solution proposed by researchers is concentration [1], which reduces the total number of network nodes by sharing each network interface among multiple terminals. A mesh network employing 4-way concentration would lead to a 4x reduction in effective node count (see figure 1(b)). Compared to the original network, a concentrated mesh has a smaller diameter and a diminished area footprint that results from improved resource sharing. While concentration is a key element in the design of scalable networks, it is not sufficient by itself. Physical limitations restrict the degree of concentration, while a reduction in channel count increases the available per-channel bandwidth, potentially leading to poor wire utilization.

The most recent effort aimed at scaling on-chip interconnects uses a butterfly network mapped onto a two-dimensional substrate. The resulting topology, called the flattened butterfly, yields a two-level hierarchical organization [9]. In the 64 node network, shown in Figure 1(c), 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 abundant on-chip wire bandwidth by spreading it over multiple channels.

Unfortunately, the flattened butterfly is not truly scalable, as the channel count in each dimension grows quadratically with the number of nodes. 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, it requires a complicated routing and buffer reservation scheme, potentially leading to additional energy expanded per packet [9].

3 Generalized Express Channels Framework

Planar silicon technologies are best matched to two-dimensional networks. While topologies with higher dimensionality can be embedded in silicon through flattening, such an embedding can result in awkward network layout leading to non-minimal channel lengths and long wire delays. Flattened topologies are acceptable as long as all channels are manhattan-minimal, meaning they use the shortest manhattan routes between any two points. In addition to increased channel traversal times, channels of non-minimal length complicate design routability and adversely impact network power consumption.

We anticipate that desirable NOC topologies have low dimensionality and instead rely on express channels [2]

for improved connectivity and latency reduction. The flattened butterfly, for instance, can be viewed as a concentrated mesh with express links connecting every node with all non-neighboring routers along the two dimensions.

Another observation is that the communication fabric need not be restricted to point-to-point links or all-to-all crossbars. An attractive middle ground might be attained by making multiple destination nodes accessible from a single channel in a one-to-many configuration. We explore this model of connectivity in later parts of the paper.

Because of the enormously large space of possible topologies, expressing an arbitrary degree of connectivity requires enumerating the set of destination nodes accessible via each channel from a given router. However, a reasonable simplification would be to consider a network where the connectivity is the same for each node, subject to network boundaries (i.e., no link or destination lies off the edge of the network). Building on the k-ary n-cube model of connectivity, we define the five-tuple <n, k, c, o, d> as:

n - network dimensionality
k - network radix (nodes/dimension)
c - concentration factor (1 = none)
o - maximum out-degree (output channels/node/direction)
d - per-channel connectivity (destinations/channel)

The first three parameters are self-explanatory, but the last two are worth elaborating. The node's out-degree, o, is expressed as the *maximum* number of outputs in a given direction[1]. In a mesh, the out-degree is one; in a flattened butterfly in Figure 1(c), it is three, since that's the maximum number of outputs per direction, corresponding to the edge routers. Given the out-degree and the concentration factor, the upper bound on router radix can be computed as $4o + c$, although the actual radix could be smaller, since any link whose first destination lies outside the edge of the network is obviously omitted. For instance, the true radix of the flattened butterfly in Figure 1(c) is 10, since every node has exactly six network and four local ports.

The last parameter, per-channel connectivity, is equal to one in a mesh and the flattened butterfly. However, as alluded to earlier, multiple destinations can be connected to a single channel, increasing the value of d.

Using the proposed taxonomy, the five-tuple for a 8-ary 2-cube (a 2D mesh with 8 nodes/dimension) is <2, 8, 1, 1, 1>. The same network mapped to a 4-way concentrated mesh becomes <2, 4, 4, 1, 1>. If connected via a flattened butterfly, the network is expressed as <2, 4, 4, 3, 1>.

Finally, it is worth pointing out that hierarchical networks, where each level uses the same or different topologies, can also be expressed via this taxonomy with a five-tuple per level.

[1] In two-dimensional networks, such as a mesh or a flattened butterfly, packets travel in the four cardinal directions. In one-dimensional topologies (eg: rings), only two directions are available.

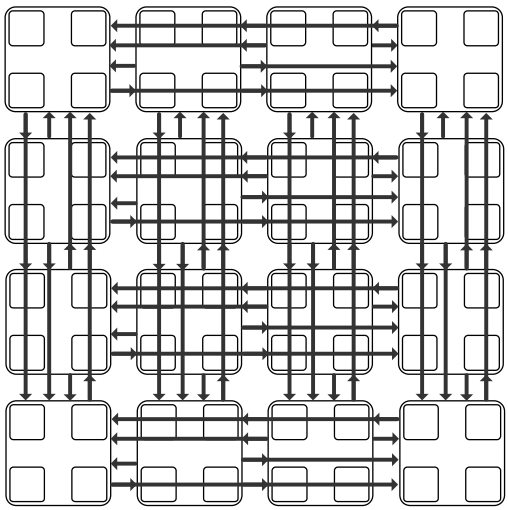


Figure 2. Multidrop Express Channels topology on a 4x4 grid with 4-way concentration (64 terminal nodes).

4 Multidrop Express Channels

The key driver behind MECS is the observation that performance and scalability in on-chip networks should be attained through judicious wire management. Minimizing the hop count is important, as intermediate routers are the source of significant delay and energy overhead. On the other hand, increasing connectivity by adding point-to-point links leads to low channel utilization, high serialization latencies, and unscalable channel count.

4.1 MECS Overview

Multidrop Express Channels are based on the notion that multiple destinations can be accessed via a single physical channel. Figure 2 shows the proposed topology in a 64-node network with 4-way concentration. Note that MECS do not require concentration; rather, the two technologies are complementary. The key characteristics of MECS are as following:

- Number of bisection channels in each dimension is equal to the network radix (nodes per dimension), k.
- The maximum hop count is two.
- Discounting the local ports, each node has at most four outputs – one per direction as in a mesh – and $2(k-1)$ inputs, akin to the flattened butterfly.

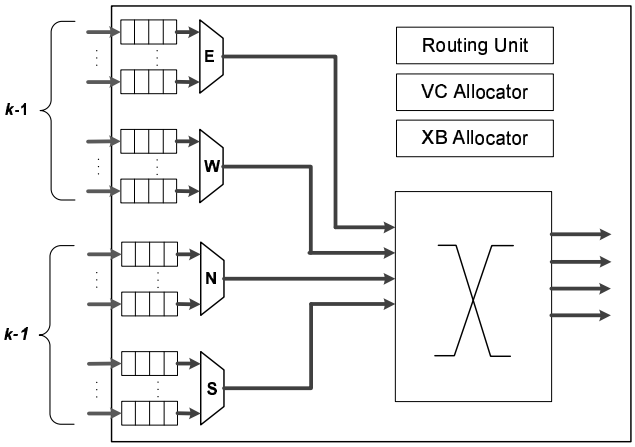


Figure 3. MECS router microarchitecture

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 hop count naturally leads to low network latencies. And 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.

4.2 Microarchitecture

Figure 3 depicts the microarchitecture of a MECS router with $2(k-1)$ network inputs and four outputs. As shown, each input port has only one virtual channel fifo, and all inputs from a given direction share a single crossbar port. This organization keeps the crossbar complexity low, minimizing its area and delay. Control complexity is comparable to a conventional mesh router with an equivalent number of virtual channels.

Because MECS make use of long wires, they require repeaters to minimize signal transmission times and reduce the capacitative load due to multiple receivers. And since signals (flits) should not be propagated further than their destinations, repeaters can be augmented with some simple logic to decide whether to transmit a flit. Once a flit reaches its destination, the intelligent repeaters will not propagate it farther, thus saving channel power.

4.3 Analysis

Table 1 compares the concentrated mesh (CMesh), flattened butterfly and MECS topologies on a number of parameters. For each topology, the first column (highlighted in gray) provides analytical expressions for computing parameter values. The formulas are expressed in terms of the network radix (k), channel bandwidth (B), and the degree of concentration (c). Note that the bandwidth is specified in terms of the width of a single channel in a concentrated mesh. The second column for each topology quantifies the parameters for a 4-ary mesh with 4-way concentration, and the third column repeats the analysis for a 8-ary mesh also with 4-way concentration.

A few trends are worth highlighting:

- The maximum hop count in a CMesh grows proportionately with mesh size, while staying the same in both MECS and flattened butterfly topologies.
- As the network radix doubles from 4 to 8 nodes per dimension, the number of bisection MECS channels in each direction also doubles from 2 to 4, while in the flattened butterfly it quadruples from 4 to 16.
- The crossbar area complexity, computed as $(router\ ports \cdot \frac{BW}{port})^2$, is the highest for the CMesh router and the lowest in the flattened butterfly. The result appears counter-intuitive, since the routers in the flattened butterfly have significantly more ports than those in other topologies. But because the per-port bandwidth in the flattened butterfly is only a fraction of the bisection bandwidth and the port count is a fraction of the bisection channel count, the area cost of the crossbar ends up being small. MECS topologies have considerably higher per-channel bandwidth than the flattened butterfly, but since the number of crossbar ports in MECS routers is low, the total area is just slightly higher than that in the flattened butterfly and significantly lower than in a CMesh.
- Estimating the buffer requirements requires knowing the number of VCs per port, α, and the depth of each VC, β. We have reasonably assumed that the concentrated mesh requires a relatively high number of VCs to avoid head-of-line blocking [1]. On the other hand, both the flattened butterfly and MECS topologies have only one VC per port, mitigating the adverse effects of head-of-line blocking through multiple ports.

  The depth of each VC, β, is a bit more difficult to estimate. At a minimum, enough buffers must be provided to cover the round-trip credit time, which is affected by the router microarchitecture and wire delay. Additional buffering can improve the throughput of the network. In any case, both the flattened butterfly and MECS topologies are likely to require greater VC depth than the CMesh to cover the wire delays associated with longer channels.

  Under our assumptions, the flattened butterfly requires less buffer space (in bits) than the MECS topology. The reason is that only a fraction of the bisection bandwidth reaches each router in the flattened butterfly due

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

	CMesh			Flattened Butterfly			MECS		
k – network radix c – concentration α – VCs per port		k=4 c=4 α = 8	k=8 c=4 α = 8		k=4 c=4 α = 1	k=8 c=4 α = 1		k=4 c=4 α = 1	k=8 c=4 α = 1
Max hop count	$2k-1$	7	15	2	2	2	2	2	2
Bisection channels (per direction)	1	1	1	$\left(\frac{k}{2}\right)^2 = \frac{k^2}{4}$	4	16	$\frac{k}{2}$	2	4
BW/channel	B	B	B	$\frac{4B}{k^2}$	$\frac{B}{4}$	$\frac{B}{16}$	$\frac{2B}{k}$	$\frac{B}{2}$	$\frac{B}{4}$
Node out-degree (o) (per direction)	1	1	1	$k-1$	3	7	1	1	1
Router radix $(4o+c)$	$4+c$	8	8	$4(k-1)+c$	10	18	$4+c$	8	8
Crossbar complexity $\left(radix \cdot \frac{BW}{channel}\right)^2$	$[(4+c)B]^2$	$64B^2$	$64B^2$	$\left[(4(k-1)+c)\frac{4B}{k^2}\right]^2$	$\sim 6B^2$	$\sim B^2$	$\left[(4+c)\frac{2B}{k}\right]^2$	$\sim 9B^2$	$\sim 2B^2$
Node in-degree (i) (per direction)	1	1	1	$k-1$	3	7	$k-1$	3	7
Buffer requirements $4i\left(\frac{BW}{channel}\right)\alpha\beta$ β – Buffers/VC	$4\alpha\beta$	$32B\beta$	$32B\beta$	$4(k-1)\left(\frac{4B}{k^2}\right)\alpha\beta$	$1.5B\beta$	$\sim 1B\beta$	$4(k-1)\left(\frac{2B}{k}\right)\alpha\beta$	$3B\beta$	$3.5B\beta$

to the high degree of channel partitioning. Furthermore, as the network is scaled to a larger number of nodes, the amount of per-channel bandwidth decreases exponentially, while the number of input ports grows linearly. As a result, the buffer requirements decrease proportionately to the reduction in bandwidth per router. In contrast, the amount of bandwidth reaching each router remains nearly constant in the MECS network, so the buffer requirements also remain approximately the same.

4.4 Extensions

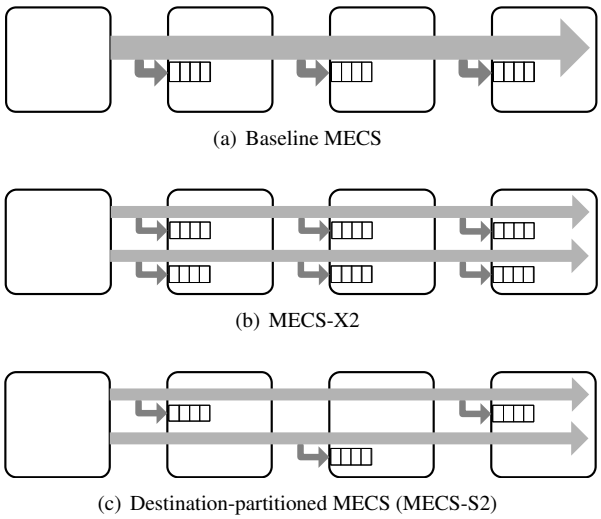
A good topology seeks to minimize the packet latency while maximizing bandwidth utilization. Reducing the latency through improved connectivity requires balancing the channel count against available bandwidth. Too many narrow channels – the serialization latency dominates; few wide channels – intra-channel bandwidth may be wasted. The latter can occur, for instance, if the width of a channel exceeds the size of a frequently-occurring packet type, such as a short read request packet or a coherence transaction.

In wire-rich on-chip networks, Multidrop Express Channels as presented thus far (Figure 4(a)) may suffer from the second problem – not utilizing all of the channel bandwidth in some cases. The flattened butterfly is less susceptible to this problem, since it partitions the bandwidth among more channels, yielding less bandwidth per channel. For MECS, the obvious solution is to partition each channel into multiple ones.

One option, shown in Figure 4(b), is to simply divide each original MECS into two or more channels, each with an equal fraction of the original bandwidth and same degree of connectivity. This configuration, called MECS-X2, would be expected to improve bandwidth utilization and reduce head-of-line blocking, both of which should improve throughput. Latency at low loads, however, might suffer due to an increase in the serialization delay.

A possible variant of this scheme would completely replicate the networks, such that each network has full connectivity of the original but with a fraction of the bandwidth. An advantage of such a design is that it minimizes the number of input ports per router, keeping the arbitration logic fast, while also reducing the combined crossbar area.

A completely different option aimed specifically at cost reduction is to again partition each MECS into two or more channels, but this time to interleave the set of destination nodes among the resulting links. If the destination sets of the partitioned MECS are mutually exclusive, than the number of inputs at each router remains the same as in the original topology, while the amount of buffering required is reduced by the partitioning factor. Figure 4(c) shows the resulting configuration, which we call *destination-partitioned* MECS. Note that by taking this idea to extreme and partitioning each original MEC such that every destination gets its own channel, we end up with the flattened butter-



(a) Baseline MECS

(b) MECS-X2

(c) Destination-partitioned MECS (MECS-S2)

Figure 4. MECS variants for cost-performance trade-off. Only one dimension is shown for simplicity.

fly topology. We will evaluate the performance of this and MECS-X2 configurations, along with other topologies, in Section 5.

4.5 Related Work

Multidrop Express Channels bear some resemblance to conventional multi-drop (broadcast) buses. The key difference is that a bus is an all-to-all medium, whereas MECS are a one-to-all paradigm.

A topology similar to MECS was used inside the YARC router [14], which used an 8x8 grid of switches to implement a radix-64 router. The switches were connected by private MECS-like buses in rows and point-to-point channels (similar to a flattened butterfly) in columns. The key difference in our proposed topology is the use of uni-directional one-to-all channels in both dimensions, which gives MECS desirable performance and scalability properties.

Kim et.al. proposed to extend the flattened butterfly topology through the use of bypass links, which allow flits to use non-minimal paths [9]. The notion of bypassing is similar to our Multidrop Express Channels; however, its use in the flattened butterfly network requires a complex reservation protocol as input ports are shared between multiple channels. MECS do not need special routing, have dedicated input ports, and require significantly fewer channels.

Table 2. Simulated configurations.

	64 nodes	**256 nodes**
Traffic patterns	bit complement, uniform random, self-similar, transpose	
Traffic type	64- and 576-bit packets	
Topology rows x columns x concentr	8x8x1 Mesh 4x4x4 CMesh, CMesh-X2 4x4x4 FBfly 4x4x4 MECS, MECS-X2	16x16x1 Mesh 8x8x4 CMesh, CMesh-X2 8x8x4 FBfly, FBfly4 8x8x4 MECS, MECS-X2, MECS-S2
Bisection Bandwidth	Mesh: 256 bits/channel CMesh: 512 bits/channel FBfly, MECS: 512 bits (split among channels)	Mesh: 512 bits/channel CMesh: 1024 bits/channel FBfly, MECS: 1024 bits (split among channels)
Router latency	Mesh: 2 cycles CMesh: 3 cycles FBfly, MECS: 3 cycles	Mesh: 2 cycles CMesh: 3 cycles FBfly, MECS: 4 cycles
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

5 Evaluation

5.1 Methodology

To compare the different topologies, we used a cycle-precise simulator that models all router pipeline delays and wire latencies. We compared the mesh, concentrated mesh (CMesh), and flattened butterfly (FBfly) topologies to MECS on two network sizes: 64 nodes and 256 nodes. Except for the mesh, all topologies use 4-way concentration, reducing the effective node count to 16 and 64, respectively. Table 2 summarizes the simulated configurations.

We modeled a bimodal packet size distribution, consisting of short 64-bit packets and long 576-bit packets. The bisection bandwidth across all topologies was kept constant. Thus, the concentrated mesh has twice the per-channel bandwidth as the basic mesh, while the flattened butterfly and MECS topologies evenly distribute this bandwidth among their links.

All of our simulated networks employ dimension-order routing (DOR). Packets in topologies with express channels (flattened butterfly and MECS) always choose the longest link that minimizes the distance to the destination (i.e., bypasses the greatest number of intermediate routers). Overshooting the destination is disallowed, as backtracking creates cycles in the network graph and can lead to deadlock.

We assume a router latency of 2 cycles in a mesh and 3 cycles in a CMesh, regardless of network size. For the smaller network, both FBfly and MECS topologies have router latencies of 3 cycles, which increase to 4 cycles in the larger network. The additional latency is intended to cover the longer arbitration delays associated with the increase in port count.

For the workloads, we used a synthetic mix consisting of bit complement, uniform random, self-similar and transpose traffic permutations. The self-similar pattern models bursty traffic and uses a randomly generated fractional Gaussian noise distribution with a Hurst constant value of 0.8 [5].

5.2 Results: 64 nodes

For the smaller network, we simulated the mesh, CMesh, CMesh-X2, flattened butterfly, MECS and MECS-X2 topologies. Both the CMesh-X2 and MECS-X2 (Figure 4(b)) partition every baseline channel into two, each with the same connectivity and half of the bandwidth as the original. In the basic concentrated mesh, the wide channels contribute to under-utilization of the available wires. As a result, the CMesh-X2 topology sacrifices very little in terms of zero-load latency while delivering substantially higher throughput than the basic CMesh. Hence, we only present results for CMesh-X2.

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%. Baseline MECS topology has consistently 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 is able to improve on that by reducing the hop count, easily amortizing the increased router latency. At low loads, it also benefits from wider channels due to a reduction in the serialization delay of large packets. The flattened butterfly and MECS-X2 have the same degree of connectivity, same number of bisection channels, and same bandwidth per channel; as such, it is expected that 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 is different when one considers the throughput of different topologies. The mesh, due to its high degree of pipelining, yields the highest throughput on three of the four workloads. CMesh-X2 has comparable performance, due to a combination of low hop count and wide channels. The flattened butterfly, on the other hand, has the lowest throughput on three traffic patterns as it cannot effectively utilize all of the available channels. MECS and MECS-X2 fall in the middle, although the latter enjoys higher throughput than the basic MECS on all of the permutations, and the highest throughput of any topology on the bit-complement pattern.

The transpose traffic pattern deserves a separate look, as the flattened butterfly achieves considerably higher throughput on it than either MECS variant. This permutation mimics a matrix transpose operation, whereby all nodes from a given row send messages to the same column. This happens to be a particularly favorable permutation for the FBfly topology, as packets from different routers in each row arrive at the "corner" node before changing dimensions and routing to their destinations via dedicated point-to-point links. As a result, there is never any interference between packets at the crucial corner router. In MECS topologies, on the other hand, packets arrive at the turn node via separate channels but then serialize on the shared outbound link. We believe that such pathological cases can be avoid through improved routing policies, and leave it to future work to validate this hypothesis.

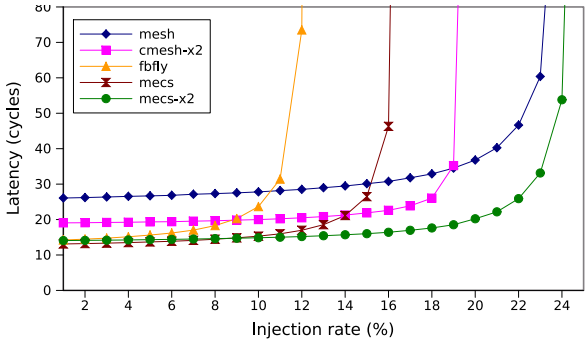
5.3 Results: 256 nodes

As we scale the topology to 256 nodes (64 with concentration), we double the per-channel bandwidth in the mesh and the CMesh. We also double the number of bisection channels in the MECS topology, since the total number of nodes per dimension doubles. As a result, the bandwidth per MECS channel remains the same. On the other hand, FBfly quadruples its channel count, consequently experiencing a 2x reduction in per-channel bandwidth.

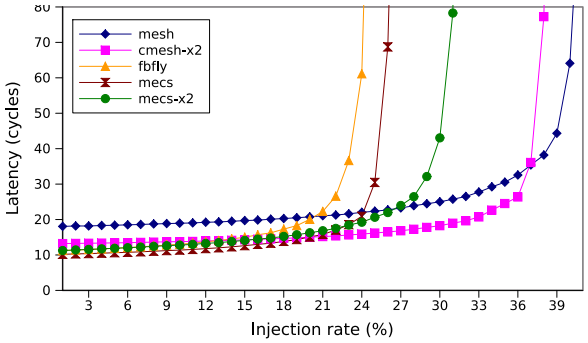
To combat FBfly's channel explosion, we also simulate a flattened butterfly with reduced connectivity. Instead of providing full connectivity in each dimension, the *FBfly4* topology links every node to at most four of its nearest neighbors in each direction. As a result, traversing each 8-ary dimension requires at most two hops, for a maximum network diameter of four. FBfly4 enjoys a 3-cycle router latency, whereas both MECS and full FBfly routers have their delays increased to four cycles to account for an increase in router radix.

Finally, we experiment with destination-partitioned MECS (Section 4.4) to evaluate the effects of this cost-reduced topology on performance. This configuration, called *MECS-S2*, partitions each original MEC into two and connects half of the original destinations to each resulting channel, as shown in Figure 4(c).

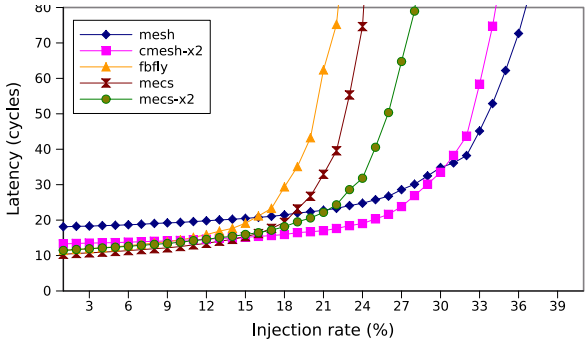
In the larger network, the basic mesh becomes decidedly unappealing at all but the highest injection rates due to its enormous zero-load latencies. The CMesh also sees its latency rise significantly, exceeding that of the flattened butterfly and MECS by 33-81% at low injection rates. In both the mesh and the CMesh, the degradation is due to the large increase in the average hop count. As expected, all MECS variants enjoy the lowest latency at low loads due to a good balance of connectivity, channel count and channel band-



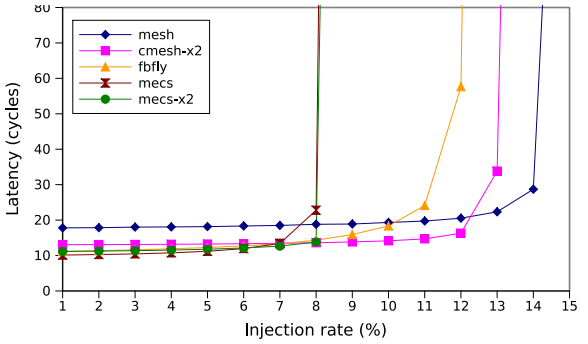
(a) Bit Complement Traffic



(b) Uniform Random Traffic

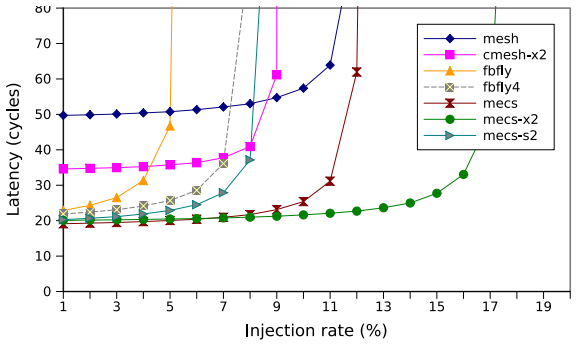


(c) Self-Similar Traffic

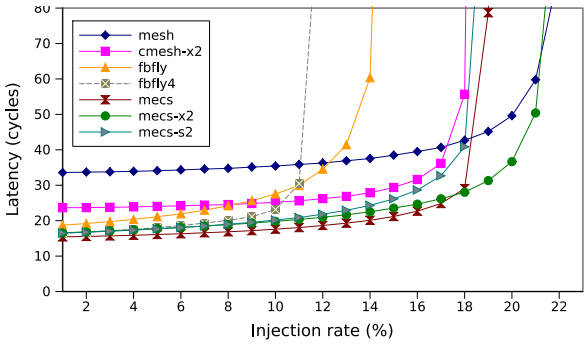


(d) Transpose Traffic

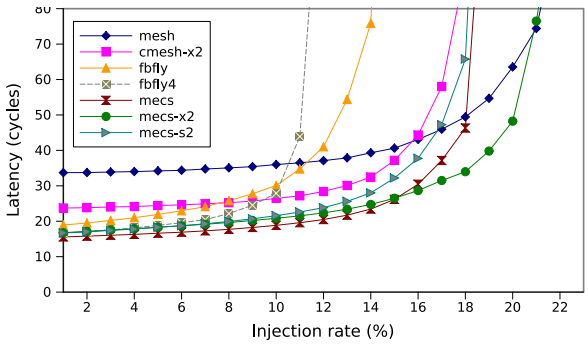
Figure 5. Load-latency graphs for a 64-node mesh, CMesh, flattened butterfly and MECS topologies.



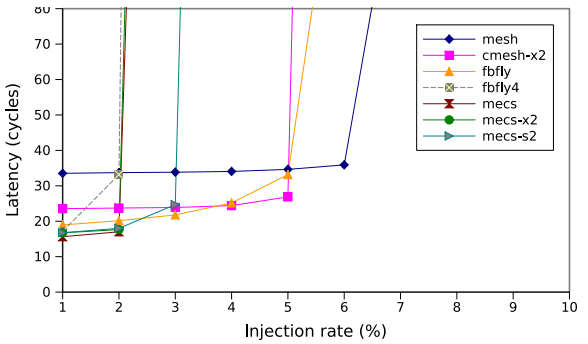
(a) Bit Complement Traffic



(b) Uniform Random Traffic



(c) Self-Similar Traffic



(d) Transpose Traffic

Figure 6. Load-latency graphs for a 256-node mesh, CMesh, flattened butterfly and MECS topologies.

width. As such, they outperform FBfly by 11-18% in terms of latency.

MECS-X2 and the mesh show the highest degree of scalability in terms of throughput. Baseline MECS also outperforms other concentrated topologies on three workloads, transpose once again being the exception as discussed earlier. MECS-S2 has a reasonable level of performance in terms of both latency and throughput, although a more detailed analysis is required to determine whether or not it is cost-efficient. 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 bandwidth. And while FBfly4 has lower latency than the basic flattened butterfly at low loads, it has the worst throughput among all topologies on three of the benchmarks.

6 Conclusion

In this work, we have sought to establish a broad framework for expressing the space of implementable and attractive topologies in on-chip networks. We observed that planar silicon restricts the effective network dimensionality to two, and that embedding networks with a larger dimension count requires flattening of the interconnect, which can lead to non-minimal channel lengths and long wire delays. Thus, we argue that the best way to improve connectivity in NOCs is through the use of express channels. Our model, called Generalized Express Channels, formalizes the notion.

Our observations motivate a new topology, called Multidrop Express Channels, that uses a one-to-many communication model to provide connectivity to multiple network nodes via a single channel. The resulting network enjoys a high-degree of inter-node connectivity, low hop count, and bisection channel count that is proportional to the arity of the dimension. Analytically, we show that MECS are competitive with other topologies in terms of cost. Compared to the previously proposed flattened butterfly topology, which fully connects routers in each dimension via dedicated point-to-point links, MECS require considerably fewer channels for the same degree of connectivity. As a result, a MECS-based 64-node network with 4-way concentration (256 total terminals) has a 17% latency advantage, on average, over the flattened butterfly (39% over CMesh-X2) on all of the workloads, and a throughput advantage exceeding 33% (5-30% over CMesh-X2) on three of the four traffic patterns.

We further demonstrated the importance of efficient wire utilization by introducing the MECS-X2 topology, which splits the bandwidth of each original MECS channel among two links, each with half of the bandwidth but same connectivity as the original. Despite a small increase in zero-load latency, MECS-X2 was shown to deliver additional throughput gains over the baseline MECS configuration. In fact, it matched or exceeded the throughput of all other topologies on three of the four workloads in the 256-terminal network.

Currently, we are evaluating the performance of MECS-based topologies on the Splash benchmark suite. In the future, we plan on performing an energy and power analysis of various MECS variants and compare their energy-efficiency to previously proposed topologies. We also plan to evaluate the efficacy of various routing policies on the performance of Multidrop Express Channels. Prior work has shown that route choice plays an important role in network performance[6, 9], potentially allowing us to overcome the throughput degradation observed on adversarial patterns such as transpose.

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, 2006.

[2] W. Dally. Express cubes: Improving the performance of k-ary n-cube interconnection networks. *IEEE Transactions on Computers*, 40(9):1016–1023, 1991.

[3] W. Dally and B. Towles. *Principles and Practices of Interconnection Networks*. Morgan Kaufmann Publishers Inc., first edition, 2004.

[4] W. J. Dally and B. Towles. Route Packets, Not Wires: On-Chip Interconnection Networks. In *International Conference on Design Automation*, pages 684–689, 2001.

[5] T. Dieker. Simulation of fractional Brownian motion . Master's thesis, University of Twente, The Netherlands, 2002.

[6] 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*, 2008.

[7] 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*, 2006.

[8] 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, 2007.

[9] J. Kim, J. Balfour, and W. Dally. Flattened butterfly topology for on-chip networks. In *MICRO '07: Proceedings of the 40th Annual IEEE/ACM International Symposium on Microarchitecture*, pages 172–182, Washington, DC, USA, 2007. IEEE Computer Society.

[10] 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, 2007.

[11] 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, 2004.

[12] 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, 2001.

[13] D. Pham, T. Aipperspach, D. Boerstler, M. Bolliger, R. Chaudhry, D. Cox, P. Harvey, P. Harvey, H. Hofstee, C. Johns, J. Kahle, A. Kameyama, J. Keaty, Y. Masubuchi, M. Pham, J. Pille, S. Posluszny, M. Riley, D. Stasiak, M. Suzuoki, O. Takahashi, J. Warnock, S. Weitzel, D. Wendel, and K. Yazawa. 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.

[14] S. Scott, D. Abts, J. Kim, and W. J. Dally. The blackwidow high-radix clos network. In *ISCA '06: Proceedings of the 33rd annual international symposium on Computer Architecture*, pages 16–28, Washington, DC, USA, 2006. IEEE Computer Society.

[15] S. Vangal, J. Howard, G. Ruhl, S. Dighe, H. Wilson, J. Tschanz, D. Finan, P. Iyer, A. Singh, T. Jacob, S. Jain, S. Venkataraman, Y. Hoskote, and N. Borkar. An 80-Tile 1.28 TFLOPS Network-on-Chip in 65nm CMOS. In *IEEE International Solid-State Circuits Conference*, February 2007.

[16] E. Waingold, M. Taylor, D. Srikrishna, V. Sarkar, W. Lee, V. Lee, J. Kim, M. Frank, P. Finch, R. Barua, J. Babb, S. Amarasinghe, and A. Agarwal. Baring It All to Software: RAW Machines. *IEEE Computer*, 30(9):86–93, September 1997.

[17] H. Wang, L.-S. Peh, and S. Malik. Power-driven design of router microarchitectures in on-chip networks. In *MICRO 36: Proceedings of the 36th annual IEEE/ACM International Symposium on Microarchitecture*, page 105, Washington, DC, USA, 2003. IEEE Computer Society.

[18] D. Wentzlaff, P. Griffin, H. Hoffmann, L. Bao, B. Edwards, C. Ramey, M. Mattina, C.-C. Miao, J. Brown, and A. Agarwal. On-chip interconnection architecture of the tile processor. *Micro, IEEE*, 27(5):15–31, Sept.-Oct. 2007.