Scaling Towards Kilo-Core Processors with Asymmetric High-Radix Topologies

Nilmini Abeyratne, Reetuparna Das, Qingkun Li, Korey Sewell, Bharan Giridhar,
Ronald G. Dreslinski, David Blaauw, Trevor Mudge
Advanced Computer Architecture Laboratory, University of Michigan, Ann Arbor
{*sabeyrat,reetudas,qingkunl,ksewell,bharan,rdreslin,blaauw,tnm*}@*umich.edu*

Abstract

In this paper, we explore the challenges in scaling on-chip networks towards kilo-core processors. Current low-radix topologies optimize for fast local communication, but do not scale well to kilo-core systems because of the large number of routers required. These increase both power and hop count. In contrast, symmetric high-radix topologies optimize for global communication with fewer hop counts, but degrade local communication with their large, slow routers.

To address both local and global communication optimizations independently, we decouple the interconnect design using asymmetric high-radix topologies. By setting a design goal of matching router speed with wire speed, our proposed topologies use fast medium-radix routers to optimize for local communication and a few slow high-radix routers that reduce hop count to optimize for global communication. Our asymmetric high-radix designs are enabled by recently proposed Swizzle-Switches*, which allow us to achieve performance scalability within realistic power budgets.*

We propose and evaluate two asymmetric high-radix topologies: Super-Star *(asymmetric folded Clos) and* Super-StarX *(asymmetric folded Clos with superimposed mesh). Our evaluations show that the best performing asymmetric high-radix topology improves average network latency over a mesh by* 45% *while reducing the power consumption by* 40%*. When compared to symmetric high-radix topologies network throughput is improved by* 2.9× *while still providing similar latency benefits and power efficiency.*

1. Introduction

Today's chip designers have resorted to increasing the number of cores in a chip as a power-efficient approach to throughput scaling. Processors with 10 to 100 cores [1,2,3,4,6] are already in the market today, and a processor with 1000 cores (kilo-core) may soon be a reality.While *off-chip* interconnection networks for 100s of nodes have been studied in the past, a power and performance scalable on-chip network for a kilo-core chip is a new challenge.

If we use a conventional topology constructed out of low-radix routers[1], such as a *2D-Mesh* [6, 18, 35, 42, 45], then the number of routers required increases as the number of cores increases. The power consumption of this growing number of routers coupled with the decreased performance resulting from larger hop counts will soon become prohibitive.

One solution to this problem is to consolidate routers into a few large but efficient high-radix switches. While high-radix switch designs were thought to be impractical due to the power and area complexity, recent work with the *Swizzle-Switch* design [14, 36, 37, 38, 41] has demonstrated that on-chip high-radix switches are feasible. The *Swizzle-Switch* is shown to scale up to a radix of 64 while supporting 128-bit channels, consuming less than 2W of power and operating at a frequency of 1.5GHz in 32nm technology. High-radix topologies facilitated by *Swizzle-Switches* make it possible to design scalable on-chip networks for kilo-core processors within realistic power budgets.

A high-radix switch can be utilized to improve scalability of interconnects in kilo-core chips by concentration [8], where multiple nodes share a router, thereby reducing the number of routers and network diameter. Also, high-radix switches can be used for designing a topology which provides more physical express links between non-adjacent routers [21], again reducing the network diameter. However, there are two problems with these approaches. First, using concentration to scale common designs (e.g., 2D-meshes), leads to lower network throughput because of bandwidth bottlenecks in inter-router links. Second, spatially close-by nodes are communicating through slower high-radix switches, degrading the performance of local communication. Thus, conventional high-radix topologies trade-off performance of local communication between close-by nodes for improving performance of global communication by reducing hop-count between nodes that are farther apart.

Our solution to mitigate these problems is an *asymmetric high-radix topology*. The key design principle of such topologies is to match the frequency of the routers with the length of the wires that connect them. For local communication, wires are short and hence wire delay is small. Therefore, the routers that facilitate local communication should operate at a higher frequency and lower radix to ensure that both wire and router delays are balanced and neither dominates overall latency. Since communication is local, low hop count is maintained even with lower radix routers. In contrast, global communication inherently spans long distances and hence incurs large wire delay. The global router can afford to be slow because the wire latency will be large at most frequencies. Thus, the router frequency can be reduced and its radix can be increased. To offset the effect of the slower router, the high radix of the global router ensures that the number of hops is

[1] Radix is defined as the number of ports in a router.

978-1-4673-5587-2/13/$31.00 ©2013 IEEE

reduced, which is important for lowering network latency for global communication.

Based on the above design principle, we propose two asymmetric high-radix topologies for kilo-core processors: *Super-Star* and *Super-StarX*. *Super-Star* is a hierarchical star topology in which a cluster of nodes are connected to a fast medium-radix local router. All local routers are connected by a high-radix global router. The network diameter is two hops. To increase network throughput we duplicate the global routers and there is no connection between the global routers. *Super-Star* with multiple global routers has the same connectivity as a folded-Clos topology [20] with one middle stage. Unlike current on-chip implementations of folded-Clos which assume equal radix routers, we explore *Super-Star* with high-radix global routers and low-radix local routers.

The second design, *Super-StarX*, extends the *Super-Star* design to permit adjacent local routers to directly communicate with each other instead of going through a global router, which further improves the performance of local communication. This optimization increases the radix of local routers by only four, which does not significantly decrease the frequency of local routers. The connections to global routers remain the same as in *Super-Star* and hence, global communication is as efficient as *Super-Star* with a network diameter of only two hops.

As a comparison point a third design, *Super-Ring*, is a hierarchical ring topology that does not follow our design principle of matching router delay with wire delay. In *Super-Ring*, a cluster of local routers is connected to a medium-radix global router. The global routers are then connected in a ring. The *Super-Ring* provides greater connectivity between global routers compared to *Super-Star* and *Super-StarX*.

We show that *Super-Star* and *Super-StarX* topologies, unlike meshes and symmetric high-radix topologies, are *energy proportional*. Their achieved throughput is proportional to the power consumed. The network throughput and power consumption can be turned up or down by varying the number of global routers. Thus network architects can choose fewer global routers at design time or power-gate the global routers at run-time. It is possible to power-gate the global routers because even a *single* global router assures *full network connectivity*.

We model a processor with 576 nodes in 15nm technology. This model provides a reasonably large system to study the scalability of interconnect topologies towards future kilo-core chips. We study the proposed network designs through detailed floor-planning, circuit-level delay analysis of routers and wires, network power models, and micro-architectural cycle accurate performance simulations. We study statistical traffic, and also 44 different benchmarks with multiprogrammed workloads of single threaded and multi-threaded shared-memory applications.

Our evaluations show that the best performing asymmetric high-radix topology improves average network latency over a mesh by 45% while reducing the power consumption by 40%. When compared to symmetric high-radix topologies (i.e. concentrated meshes and flattened butterfly), our proposed topologies improve network throughput by 2.9× while still providing similar latency benefits and power efficiency. Over a varied set of application workloads, the final proposed topology improves application performance by 17%, while reducing power consumption by 39%.

In summary, our key contributions are:

- We propose asymmetric high-radix topologies for performance and power scalable on-chip networks for designing kilo-core systems. Our proposed topologies optimize for both local and global communication.
- Our key design principle for asymmetric topologies is to match router speed with wire speed. Fast medium-radix routers support local communication along short wires and a few slow high-radix routers support global communication by reducing hop count. The global high-radix routers can afford to be slow because wire delays of global routes are inherently longer.
- Based on our design principle, we propose and evaluate two asymmetric high-radix topologies: *Super-Star* (asymmetric folded-Clos) and *Super-StarX* (asymmetric folded-Clos with superimposed mesh). These topologies vary in their degree of local and global connectivity.
- We also find that *Super-Star* and *Super-StarX* topologies, unlike meshes and symmetric high-radix topologies, are *energy proportional*.

2. Motivation and Background

2.1. Scaling of Low-Radix Mesh Topology

Low-radix mesh [6, 18, 35, 42, 45] topologies have become popular for tiled manycore processors because of their low complexity, and planar 2D-layout properties. Figure 1 shows the layout of a mesh topology. For our studies, we investigate a 576-node chip with 552 core tiles and 24 memory controller tiles. The length of a tile is 0.9*mm*. The tile dimensions are chosen such that it can accommodate a simple out-of-order ARM Cortex A15 core, 32 KB of L1 cache, 256 KB of L2 cache and a small radix-5 mesh router in 15nm design. The tiles are connected with a 24×24 2D-mesh.

Unfortunately, as we scale up the mesh topology towards kilo-core processors, it shows poor performance scalability due to its quickly growing network diameter. The large number of

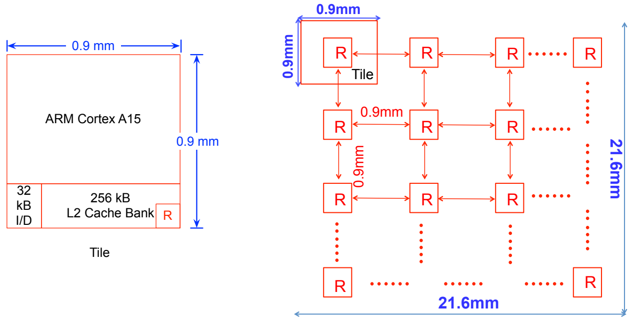

Figure 1: (a) A Tile and (b) Mesh Topology

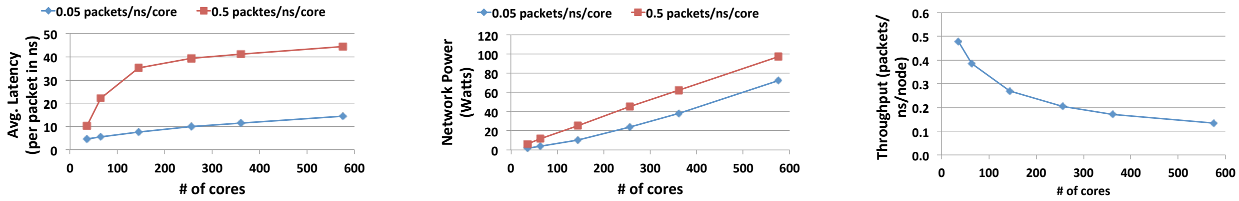

Figure 2: Scaling of mesh topology with number of cores: (a) Network latency, (b) Network power and (c) Throughput per core.

routers required by the mesh topology pushes the overall network power far beyond practical limits [11, 12]. High average hop count also leads to high variability of available per core bandwidth [26] and exacerbates worst case latency.

Figure 2 illustrates the scaling characteristics of the mesh topology as we increase the number of cores from 36 to 576 (Section 4 provides simulation and modeling details). The network latency and power is shown for two injection rates, $0.05 packets/ns/core$ (low) and $0.5 packets/ns/core$ (high). Even at a low injection rate the network latency degrades by 3× as we increase the number of cores from 36 to 576. At the high injection rate, the degradation in latency is steeper. Thus, higher performance afforded by increasing number of cores can be offset by communication overheads. Figure 2(b) shows the steep increase in network power from $6.3W$ to $97.1W$ as we increase the number of cores from 36 to 576. Figure 2(c) illustrates that the available per core throughput reduces by 3.7× as we increase the number of cores from 36 to 576. Ideally, we would like the network to provide a constant per-core bandwidth with increasing number of cores, such that the performance of individual cores is not effected by scaling up the number of cores.

The above studies motivate the need for a scalable interconnect topology. It can be seen that future manycore processors cannot afford the luxury of a low-complexity mesh topology. In this paper, we propose asymmetric high-radix topologies as a solution. Before we delve into high-radix topologies we give a brief background on the *Swizzle-Switch*, which is the key-enabler of our designs. For more details on implementation of the *Swizzle-Switch*, we refer the reader to recent prior work [14, 36, 37, 38, 41].

2.2. Enabling High-Radix Routers with *Swizzle-Switch*

The SRAM-inspired design of the *Swizzle-Switch* provides good scalability to large radices. Traditional matrix-style switches consist of a crossbar that routes data and a separate arbiter that configures the crossbar. This decoupled approach poses two hurdles to scalability: (1) the routing to and from the arbiter becomes more challenging as the radix increases and (2) the arbitration logic grows more complex as the radix increases. Arbiters that need to distribute their arbitration over multiple stages incur the overhead of flip-flops to store the control flow signals. The work done by Passas [30] illustrates the difficulty of implementing a multistage arbiter for a high-radix switch. In Passas' work, a radix-128 switch is shown to have a crossbar arbiter that consumes 60% of the total crossbar area and requires three stages to do arbitration.

To overcome these limitations, the *Swizzle-Switch* combines the routing-dominated crossbar and logic-dominated arbiter by embedding the arbitration logic within the switch crosspoints. The *Swizzle-Switch* design reuses input/output buses for arbitration, producing a compact design. The arbitration is done in a single cycle by comparing priority bits that are embedded in the switching fabric. At the end of each arbitration stage, the priority bits are automatically updated by setting and re-setting appropriate priority bits to achieve least recently granted order of arbitration. To reduce power, the *Swizzle-Switch* uses SRAM-like technology with low-swing output wires and a single-ended thyristor-based sense amplifier. We studied the scalability of the *Swizzle-Switch* across a wide range of radices. Figure 3 shows the frequency and energy per bit transferred of the *Swizzle-Switch* as function of its radix. Even when the radix is increased to 64, the *Swizzle-Switch* with 128-bit channels can continue to operate at a high frequency of $1.5GHz$ while consuming less than $2W$ of power. In 32nm technology, this *Swizzle-Switch* requires $\sim 2mm^2$ of area.

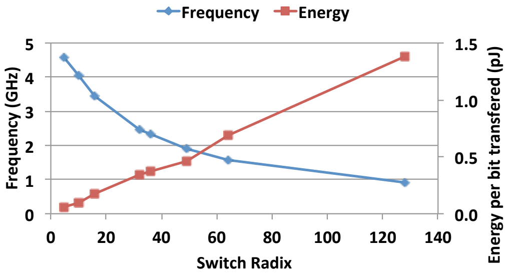

Figure 3: Scaling of a 128-bit *Swizzle-Switch* with radix.

3. High-Radix Topology Design

In this section, we explore several high-radix topologies and analyze their scalability in the context of kilo-core processors. First, we discuss symmetric high-radix topologies consisting of all equal-radix routers and their design trade-offs. Then, we discuss asymmetric high-radix topologies where router radix is guided by wire delay. These topologies are designed to optimize both local and global communication.

3.1. Symmetric High-Radix Designs

3.1.1. Concentration Balfour and Dally [8] proposed a concentrated mesh which allows a few nodes to share a router. The number of nodes sharing a router is called the concentration degree of the router. Since the router is shared, the radix of its switch increases by least its concentration degree. Concentration yields two benefits: 1) it reduces the network latency by reducing the network diameter and average hop count; and

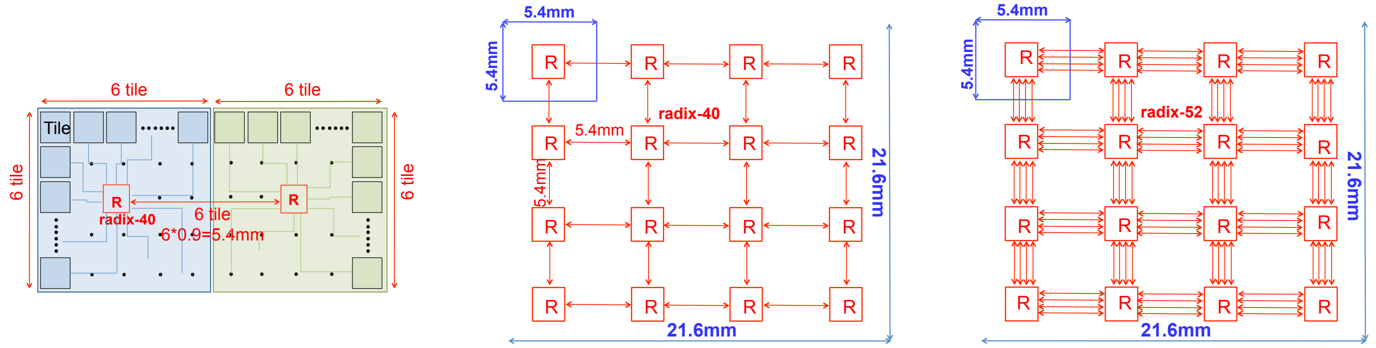

Figure 4: Concentrated Mesh Topology: (a) Layout of tiles within a cluster for a concentration degree of 36. (b) Layout of concentrated routers in a mesh. (c) Layout of concentrated mesh with four parallel links between routers.

2) it reduces the number of routers, which can lead to power savings.

However, the benefits of concentration are largely dependent on the power-frequency scalability of the switch. As we increase the concentration degree (and hence the switch radix), the routers become larger and slower in terms of frequency, and the wires which connect them become longer. Thus the benefits due to reduced hop count may be offset due to reduction in performance of individual switches.

In [8], the authors target a 64-tile system where 4 tiles share a router. We find that a concentration degree of 4 does not provide sufficient scalability for kilo-core systems. To scale to 576 nodes, we leverage *Swizzle-Switches* to increase the concentration to much higher degrees, and study the trade-offs between reduced hop count and reduced router frequency. Figure 4 (b) shows the layout of concentrated mesh with a concentration degree of 36 for our target processor design. Each router services 36 tiles. A group of 36 tiles has $5.4mm$ by $5.4mm$ dimensions (Figure 4 (a)). The longest local link between the tiles and router is $2.7mm$. The links between routers are $5.4mm$ long. The radix of each router is 40 and the router operates at frequency of $2.2Ghz$. The network diameter reduces from 46 hops to 6 hops when compared to mesh.

From our studies, we find that concentrated meshes provide significantly lower throughput than mesh. This is because concentrated meshes have lower bandwidth and the inter-router links become a bottleneck. The local links between the tiles and cores seldom become the bottleneck. Thus, we consider a new concentrated mesh design which has multiple parallel links between routers to improve throughput. However, these additional links further increase the switch radix, and hence reduce the router frequency. Figure 4 (c) shows the layout of a 36-degree concentrated mesh with 4 parallel links between the routers. The radix of each router increases to 52 and its frequency reduces to $1.8Ghz$.

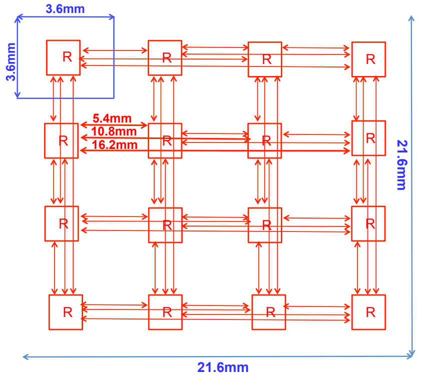

Figure 5: Flattened Butterfly Topology

In our evaluations, we show that the conflicting trade-offs discussed above limit the benefits of concentration.

3.1.2. Flattened Butterfly The flattened butterfly is a cost-efficient topology that can be extended to high-radix routers [21]. It is derived by combining the routers in each stage of a conventional multi-stage butterfly network. The flattened butterfly reduces hop count over conventional mesh by concentration as well as rich connectivity by using longer express links between non-adjacent routers.

The flattened butterfly topology can be scaled up by either increasing concentration, or increasing the dimensions (i.e. stages). For our studies, we choose to increase concentration. We limit ourselves to 2-dimensional flattened butterfly to reduce the stages and hence achieve a low network diameter of 2 hops. Also, the 2-dimensional flattened butterfly renders well to a 2D-planar layout.

The flattened butterfly uses symmetric high-radix routers, concentration, and express channels to improve scalability. Its symmetric nature trades off efficiency of local communication to achieve faster global communication. Also, its scalability in terms of network throughput is limited due to concentration.

Figure 5 shows the layout of the 4-ary 3-flat 2-dimensional flattened butterfly used in our studies. Each router is shared by 36 tiles. The cluster of tiles around a router will be similar to Figure 4(a). There are 16 routers of radix-42 operating at a speed of $2.1Ghz$. The longest link in the topology is about $17.6mm$ and is pipelined to deliver flits in 3 cycles.

3.2. Asymmetric High-Radix Designs

Above, we observed that traditional symmetric high-radix topologies trade-off local communication for global communication. These topologies have large high-radix routers which reduce hop count and optimize for global communication delay. But this is at the cost of higher local communication delay,

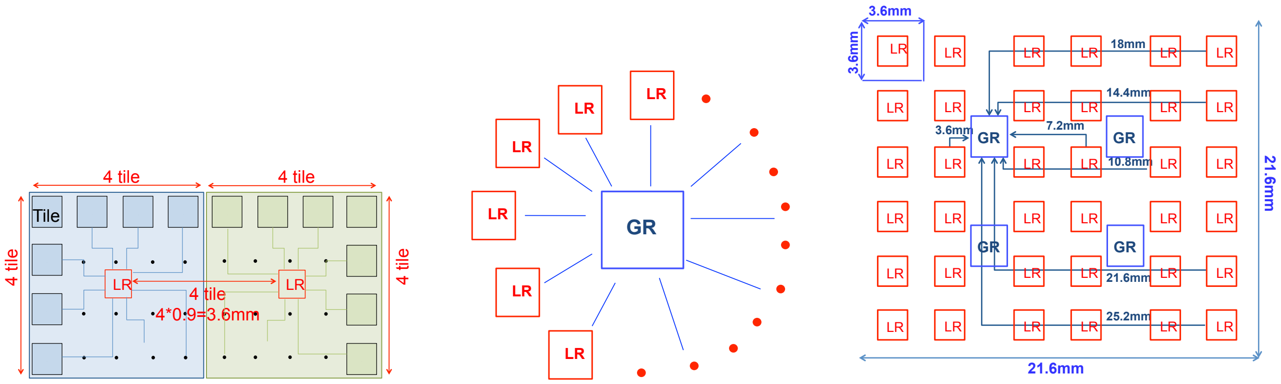

Figure 6: *Super-Star* Topology: (a) Layout of tiles within a cluster with a Local Router (LR). (b) Logical view of *Super-Star* showing connectivity between Local Routers (LR) and Global Router (GR). (c) Layout of *Super-Star* with four GRs.

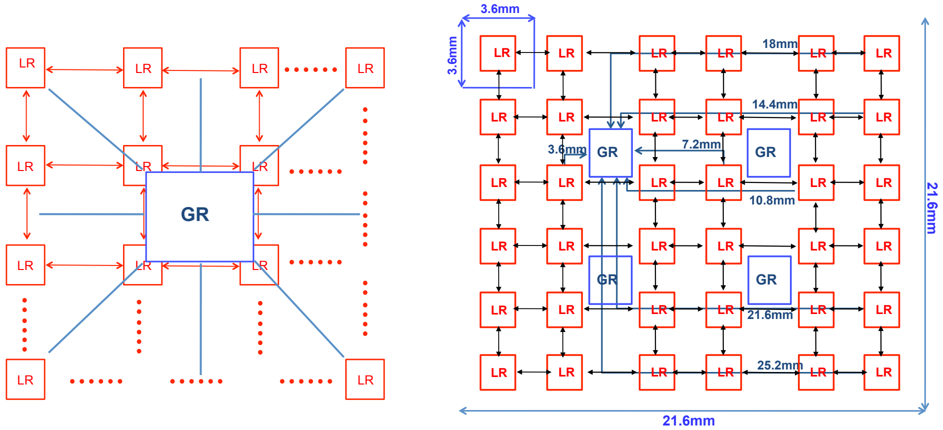

Figure 7: *Super-StarX* Topology: (a) Logical view of *Super-StarX* showing connectivity between Local Routers (LR) and Global Router (GR). (b) Layout of *Super-StarX* with four GRs.

which requires routing through the slow high-radix routers even for close-by cores.

Our approach towards designing a high-radix topology consists of three key elements. *First*, we split the communication into local traffic between cores which are near-by and global traffic between cores that are spread apart. This is not a new concept and has been used in prior interconnect designs [13] and in other contexts, such as road systems in cities, power supply grids, etc. *Second*, we make the key observation that for each type of communication, router speed should match wire speed. For local communication—where cores are close-by, wires are short, and wire delay is small—the router should be fast and have lower radix. Since communication is local, the lower radix does not increase hop count significantly. For global communication the routes will be inherently long and wire latency will be large regardless of the number of pipeline stages. Hence, global routers can afford to be slower allowing their radix to be increased. With higher radix, the number of hops is reduced, which results in lower network latency for global communication. *Finally*, we tackle the problem of reduced network throughput in highly concentrated topologies by replicating the global routers.

Based on the above guidelines we explore two high-radix topologies: *Super-Star* and *Super-StarX*. As a comparison point we also consider a third asymmetric high-radix topology that does not follow our design principle, *Super-Ring*, which employs the popular ring interconnect for global routers.

Multi-stage topologies such as trees [8, 27] and Clos [20] that have been proposed for on-chip networks have hop-counts proportional to the number of stages. The scalability of *Swizzle-Switch* to higher radices enables us to achieve optimal performance and power with only two-stages, thus precluding the need to explore greater than two-stage switches.

3.2.1. Super-Star The first asymmetric design is a hierarchical star topology. In *Super-Star*, a cluster of nodes are connected to a fast medium-radix local router as shown in Figure 6(a). Figure 6(b) shows the logical sketch of *Super-Star* with local routers and a global router. The global router is connected to all local routers. The network diameter is two hops. The number of global routers can be increased to provide higher throughput. There is no connection between the global routers. With multiple global routers, the *Super-Star* topology has the same topology connections as a 3-stage folded-Clos. However, current on-chip implementations of folded-Clos use equal radix routers [20]. This work is different in that we use few high-radix global routers and many low-radix local routers.

Figure 6(c) shows the physical layout of the *Super-Star* topology with 4 global radix-36 routers and 36 local radix-20

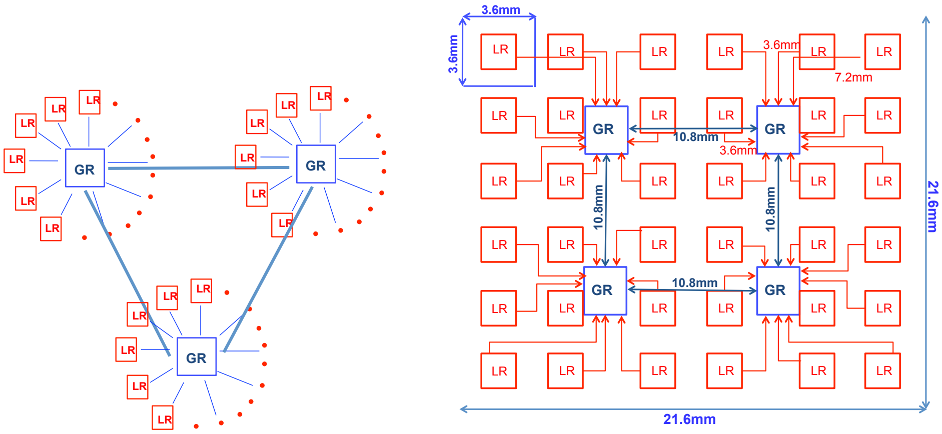

Figure 8: *Super-Ring* Topology: (a) Logical view of *Super-Ring* showing connectivity between Local Routers (LR) and Global Router (GRs). (b) Layout of *Super-Ring* with four GRs.

routers. The figure shows only a few distinct links with their dimensions for clarity. All outgoing links are pipelined to match the clock frequency of the router. Note, some global routers are spatially closer to a local router than others. However for simplicity and load balance, the global routers are chosen in a round-robin manner during the routing stage. More sophisticated routing schemes which account for wire-dimensions and buffer occupancy are also possible.

An interesting property of *Super-Star* is *energy proportionality*. The network throughput achieved by *Super-Star* topology and its power consumption is proportional to the number of global routers. Moreover, the entire network remains *fully connected* even with a single global router. Thus, network architects can choose to have fewer global routers, if they are power constrained. Alternatively, the network can have a sufficient number of global routers to satisfy the peak throughput requirement. But when network load is low, a subset of global routers can be power-gated. In mesh and traditional symmetric high-radix topologies, energy proportionality is hard to achieve because *all* routers need to be active to keep the entire network fully connected, even when the overall network load is low.

3.2.2. Super-StarX In the *Super-Star* topology, fast local communication is restricted to the cores within a cluster connected by the local routers. The local routers which are spatially close (i.e. neighbors in the layout) still need to communicate via a global router. We observe that providing connectivity between neighbors is cheap in terms of radix (the local router's radix only increases by 4, leading to minimal decrease in frequency), and this connectivity can reduce the latency of the local communication further. We refer to this new topology, which is derived from *Super-Star* by connecting the adjacent local routers as *Super-StarX*.

Figure 7(a) shows the logical sketch of *Super-StarX*. Note, all the beneficial characteristics of *Super-Star*, such as low latency, energy proportionality, etc, are preserved in *Super-StarX*. Although, sophisticated adaptive routing solutions are possible due to path diversity, we chose to implement a simple routing scheme in *Super-StarX*. The new links added between local routers are used only to communicate between neighboring local routers. All other inter-cluster communication between local routers is via the global routers. Thus, unlike concentrated mesh, in *Super-StarX*, *the maximum number of hops is still limited to two hops*. Figure 7(b) shows the layout of a *Super-StarX* topology with 4 global radix-36 routers and 36 local radix-24 routers (each local router is shared by 16 tiles). The link dimensions remain similar to *Super-Star* topology.

3.2.3. Super-Ring Our previous asymmetric high-radix topologies (i.e. *Super-Star* and *Super-StarX*) connect the global router to *all* local routers. The local routers are medium-radix, fast, and matched to local wire delay. The global routers are high-radix, slower, and matched to global wire delay. Finally, we explore a topology which does not follow our design philosophy. In *Super-Ring*, the chip is divided into four logical quadrants with one global router per quadrant. The local routers are still medium-radix and match local wire delay. However, global routers are also medium-radix and are only connected to a *subset* of local routers. To provide full network connectivity, global routers are connected to each other in a ring. Note, all global routers need to be active for full connectivity, thus this topology is not energy proportional. Figure 8(a) shows the logical sketch of *Super-Ring*. Figure 8(b) shows the layout of a *Super-Ring* topology with 36 local radix-17 routers (each local router is shared by 16 tiles) and 4 radix-11 global routers. The link dimensions between local and global routers are shorter than *Super-Star* topology.

4. Evaluation Methodology

4.1. Router Delay and Power Model

We analyze the power and delay of each component of a router such as, links, buffers and switch (i.e. *Swizzle-Switch*), through SPICE modeling in 32nm industrial process and scale it conservatively to 15nm technology. Our models include energy spent due to clocking and leakage energy. The *Swizzle-Switch* architecture has been validated with a fabricated and tested silicon protoype [37, 38]. We assume a 128-bit *Swizzle-Switch* for all routers in our topologies and determine its frequency and power consumption at different radices. For each router, we assume a buffering of 4 virtual channels per port and a buffer depth of 5 flits per virtual channel. The routers utilize simple dual clock I/O buffer design with independent read and write clocks (similar to [29]). We conducted buffer sensitivity studies which showed that this much of buffering was sufficient, even for topologies with long links. Our simulations model in detail the interface between routers operating at different frequency and multi-cycle links.

4.2. Link Delay and Power Model

Wire delays were determined using wire models from the design kit using SPICE modeling. Our analysis takes into account cross-coupling capacitance of neighboring wires and metal layers. For all links, we consider options that trade off energy for speed. We use different metal layers with either single or double spacing. Repeater insertion is adjusted so that repeaters are placed in the gaps between cores. The repeater placement was considered for all topologies to accurately estimate timing. On average the wire delay was found to be $66ps/mm$ and wire energy was found to be $0.07pJ/mm/bit$.

4.3. Performance Simulations

We use a cycle-accurate network-on-chip simulator for our analysis. All routers, irrespective of radix, use a two-stage microarchitecture [33]. We use simple deterministic routing algorithms, finite input buffering, wormhole switching, and virtual-channel flow control. The long links in different topologies were pipelined at the router frequency. The heterogeneity of frequency between routers was faithfully modeled. The activity factor of links, buffers and switches were collected from cycle-accurate simulations and integrated with power models to determine the network power.

We evaluate the proposed topologies with *uniform random statistical traffic* with a packet size of 512 bits (i.e. 4 flits). The datapath width is constant across all topologies and is equal to 128 bits. The network latency is reported is *nanoseconds* and the network throughput is reported in $packets/nanosecond/node$.

For applications, we use a trace-driven, cycle-accurate many-core simulator with the above network model integrated with core, cache and memory controller models. Note, all the different components are tightly integrated to create a close-loop simulation environment. For example, the cores stall on a cache miss, the dependency between different coherence messages is obeyed, and queueing delays at the cache controllers and memory controllers are modeled. Thus, we can measure the execution time for the different workloads we simulate. Table 1 provides the configuration details.

We use a set of multiprogrammed application workloads comprising scientific, commercial, and desktop applications. In total, we study 44 benchmarks, including SPEC CPU2006 benchmarks, applications from SPLASH-2 benchmark suites, and four commercial workloads traces (sap, tpcw,sjbb, sjas). The traces for SPEC CPU2006 where collected using dynamic binary instrumentation [32]. The commercial workload traces where collected over Intel servers. The traces for SPLASH-2 benchmarks were collected by running the benchmarks on *gem5* full-system simulator [9]. The details of how each multiprogrammed workload mix is derived from the different *single-threaded* and *multi-threaded* benchmarks are discussed in Section 5.3.

Table 1: Processor configuration

Cores	552 cores , 2-way out-of-order, 1 GHz frequency
L1 Caches	32 KB per-core, private, 4-way set associative, 64B block size, 2-cycle latency, split I/D caches, 32 MSHRs
L2 Caches	552 banks, 256KB per bank, shared, 16-way set associative, 64B block size, 6-cycle latency, 32 MSHRs
Main Memory	24 on-chip memory controllers with 4 DDR channels each @16GB/s, up to 16 outstanding requests per core, 80ns access latency

5. Results

5.1. Analysis with Uniform Random Statistical Traffic

We first study the benefits and limitations of concentration. Figure 9(a) shows the average network latency and Figure 9(b) shows the network throughput with varying degrees of concentration. As postulated in Section 3.1.1, concentration provides excellent latency benefits at the cost of reduced throughput. Also, the latency benefits flatten out after reaching a concentration degree of 36. Beyond this concentration degree the benefits due to reduced hop count is countered by reduced router frequency. The average network latency before saturation drops from $16.8ns$ in mesh to $8.9ns$ for concentration degree of 36 and increases back to $9.2ns$ at a concentration degree of 64.

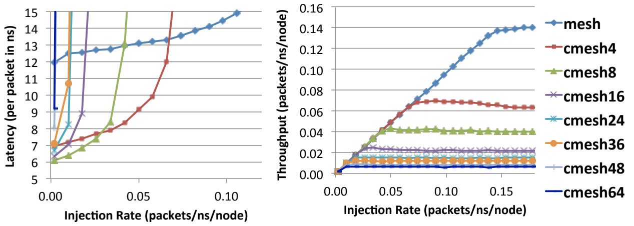

Figure 9: Network latency (a) and throughput (b) for concentrated mesh with different concentration degrees.

In order to regain the throughput lost by concentration, we experiment with a new concentrated mesh topology with multiple parallel links. For this study we choose the largest concentration degree which provides the best latency and consumes the least power i.e., concentration degree of 36. Although concentration degree of 8 has the best latency in Figure 9 (a), the higher number of routers dissipates more power. We maximize the concentration degree to reduce the number of routers and hence reduce power. Figure 10 (a) shows the average network throughput and Figure 10 (b) shows the network power as a function of achieved throughput, with varying number of inter-router links. It can be seen that although we regain some of the lost throughput by adding additional inter-router links, the power grows steeply with additional links. Each additional set of links make the router bigger (router's radix increases by 4 times the number of parallel links), slower, and increases its power. The concentrated mesh with 16 parallel links consumes a power of $100.1W$ while providing a peak throughput which is

Table 2: Router radix, link dimensions and network area for different topologies.

Topology	# Routers		Radix		Network	Avg. Link Length(mm)	
	Local	Global	Local	Global	Area	Local	Global
mesh	576	-	5	-	38.19	0.79	-
cmesh-low	144	-	8	-	13.18	1.28	-
cmesh-high	16	-	52	-	15.20	3.25	-
fbfly	16	-	42	-	10.82	3.56	-
superstar	36	8	24	36	18.24	1.80	12.90
superstarX	36	8	28	36	21.45	2.11	11.30
superring	36	4	17	11	7.12	1.80	6.48

Table 3: Bisection bandwidth wires of different topologies for equal wires study.

Non-equal # Wires		Equal # Wires	
Bus Width	Bisection Wires	Bus Width	Bisection Wires
128	6,144	512	24,576
128	9,216	256	18,432
128	10,240	256	20,480
128	10,240	256	20,480
128	36,864	73	21,024
128	44,544	64	22,272
128	9,216	256	18,432

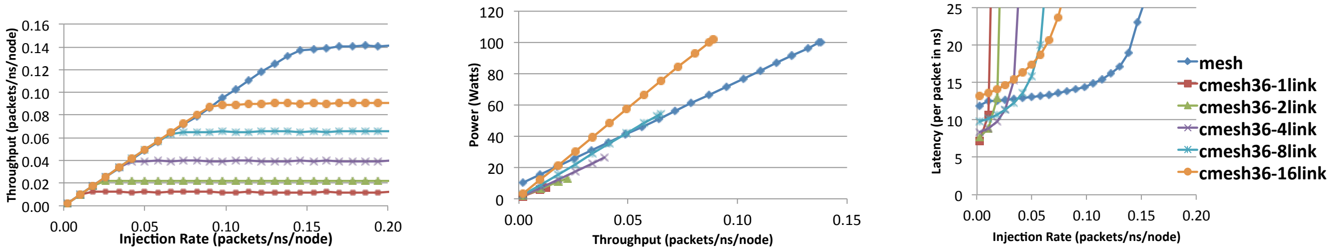

Figure 10: Network throughput (a) power (b) and latency (c) for concentrated mesh with additional number of inter-router links.

only 60% of mesh's peak throughput. Thus, we conclude that concentration alone cannot scale the interconnect to kilo-core processors.

Next, we study the different asymmetric high-radix topologies. We present the best configurations of each topology. Table 2 provides the number of routers and their radix, network area and link dimensions for different topologies. The design goal was to restrict interconnect to 5% of chip area ($466mm^2$) while meeting performance and power targets. Figure 11 shows the average network latency and network throughput for different topologies. The low-radix *mesh* topology has high average network latency because of large number of hops. However, it is also able to achieve good network throughput because it has high bandwidth. The average latency of *mesh* topology before saturation is $16.8ns$ and it saturates at the throughput of $0.14 packets/ns/node$.

In contrast, the symmetric high-radix topologies enjoy low latency because of reduced hop count. However, they quickly saturate because of bandwidth bottleneck in inter-router links. The *cmesh-low* topology has a low concentration degree of 4. The *cmesh-high* topology has a high concentration degree of 36 and in addition has 4 parallel links between the routers. The *cmesh-low*, *cmesh-high* and flattened butterfly (*fbfly*) topologies have an average network latency of $9.6ns$, $10.8ns$ and $7.9ns$ and a saturation throughput of $0.07 packets/ns/node$, $0.04 packets/ns/node$ and $0.044 packets/ns/node$. We also studied improving the bandwidth of symmetric high-radix topologies by increasing the datapath width and link width beyond 128 bits. However, we find that increasing datapath width makes the router slower as well as increases network power consumption significantly.

The asymmetric high-radix *Super-Star* and *Super-StarX* topologies enjoy both low latency and high throughput. They achieve low latency by effectively optimizing both local and global communication. They achieve high network throughput by having multiple global routers. The *Super-Star* and *Super-StarX* topologies have an average network latency of $9.3ns$ and $9.5ns$, about 45% improvement over mesh. Note, since we are simulating uniform random traffic pattern, the *Super-StarX* latency is similar to *Super-Star*. As shown later in a clustered traffic study, *Super-StarX* provides better latency for higher proportion of local traffic. Again, all average latencies are taken before saturation. While *fbfly* has a lower average latency than these topologies, it also saturates at a lower throughput. The *Super-Star* and *Super-StarX* topologies have a saturation throughput of $0.18 packets/ns/node$ and $0.20 packets/ns/node$.

The *Super-Ring* topology, although an asymmetric high-radix topology, was designed without adhering to our goal of matching router delay to wire delay. In this topology, the global routers are medium-radix, smaller and faster. Thus, global wire-delay is not matched to router speed. The local routers are still medium-radix. In addition, there is no redundancy between the global routers. Thus, inter-router links between global routers can become bandwidth bottlenecks. The *Super-Ring* provides an average latency of $10.6ns$ and quickly saturates at throughput of $0.01 packets/ns/node$. We conclude that matching wire delay with router speed is important and a naive asymmetric hierarchical topology cannot provide optimal performance.

Figure 12 shows the network power and network energy for different topologies. Figure 12 (a) plots the network power (Y-axis) as function of achieved network throughput (X-axis). The network power increases with increasing network throughput. The lines for different topologies stop at different throughputs and the end points correspond to the saturation throughput of that topology. It can be seen that symmetric high-radix topologies stop very quickly due to their lower throughput. In general, the topologies which reach further right and have a slow slope of increase in power with respect to throughput are more desirable. It can be seen that *Super-Star* and *Super-StarX* topologies achieve the best power efficiency: 1) their slope of power increase with respect to throughput is smallest and 2) their achievable throughout is farthest to the right. They can

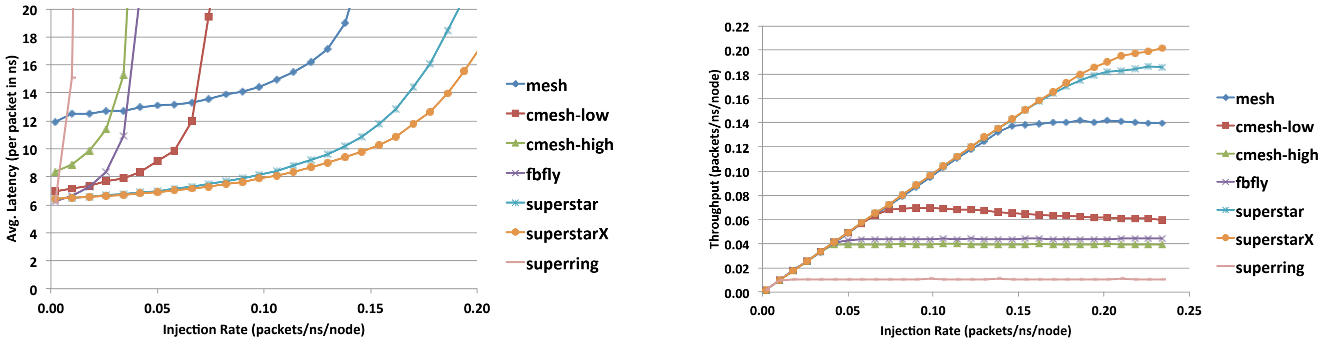

Figure 11: Performance of different topologies with uniform traffic:(a) Network latency and (b) Network throughput.

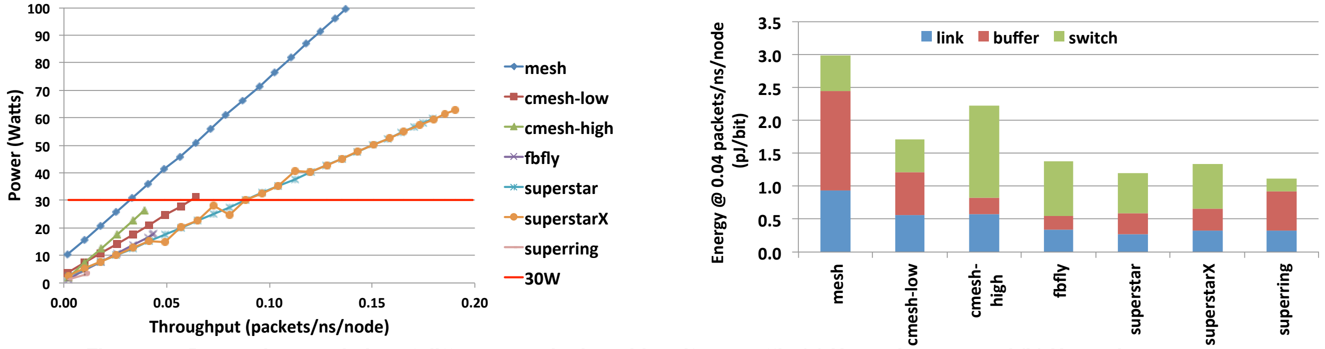

Figure 12: Power characteristics of different topologies with uniform traffic:(a) Network power and (b) Network energy.

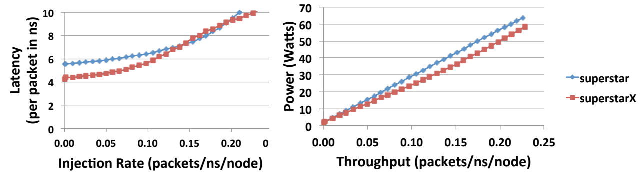

Figure 13: Network latency (a) and Network power (b) for clustered traffic study.

achieve 39% higher throughput while consuming only 60% of power when compared to *mesh*. If we limit the network power to 30*W* across all topologies, the proposed *Super-Star* and *Super-StarX* topologies can provide 3× higher throughput than *mesh* and 1.4× higher throughput than *cmesh-low*. Figure 12 (b) shows the energy per bit of the different topologies at a low injection rate of $0.04 packets/ns/node$. It can be seen that the proposed high-radix topologies trade-off link and buffer energy for switching energy.

To further emphasize the benefit of providing fast connectivity to adjacent local routers in *Super-StarX*, we simulated a clustered traffic pattern. In this traffic pattern, communication is only to cores within the same cluster or to cores in adjacent clusters. The cluster size of both *Super-Star* and *Super-StarX* is 16 tiles. The locality-aware routing policy of *Super-StarX* uses the links between local routers to route most packets. The routing policy adapts to high congestion by routing packets via the global routers when the buffer occupancy for links between local routers exceed a predetermined threshold. Figure 13 shows the network latency and network power for this study. The additional connectivity and the adaptive, locality-aware routing policy of *Super-StarX* provide much lower latency than *Super-Star* and better power efficiency.

Finally, we evaluate the energy proportionality of our proposed *Super-Star* topology. Figure 14 (a) shows the proportional growth in throughput as we increase the global routers (GRs) from 1 to 8. Figure 14 (b) shows that the network power increase with respect to achieved throughput has similar slope for all the different configurations (GR1 to GR8). Thus, if the required throughput of the system is low, designers can save power by using fewer GRs. In mesh and symmetric high-radix topologies, all routers are necessary to provide full network connectivity. Thus, it is hard to design these networks in an energy proportional manner. To bound these topologies to a lower Thermal Design Power (TDP) budget (e.g. TDP is equal to 30*W*), they will have to be either 1) under-clocked, sacrificing latency or 2) have complex source throttling mechanisms to limit the injection rates at source nodes such that the network power does not exceed the pre-decided TDP.

5.2. Bisection Bandwidth Wires

Bisection bandwidth of the topologies in Figure 11 varies significantly. We assumed a constant 128-bit datapath width, which results in different number of wires at the bisection for different topologies, as listed in Table 3. The bisection wires include wires from tiles to local routers as well as inter-router links. For a better comparison of topologies, we conducted a new study with an equal number of wires for all topologies. To achieve the same number of wires (approximately 21,000), the datapath width was adjusted according to number of links at bisection. The new channel widths are listed in Table 3. Figure 15 shows the average network latency and network power for this study. The wider datapath of mesh and *cmesh-low* causes the frequency of the router to decrease, thus we observe a small increase in latency compared to Figure 11. Except for

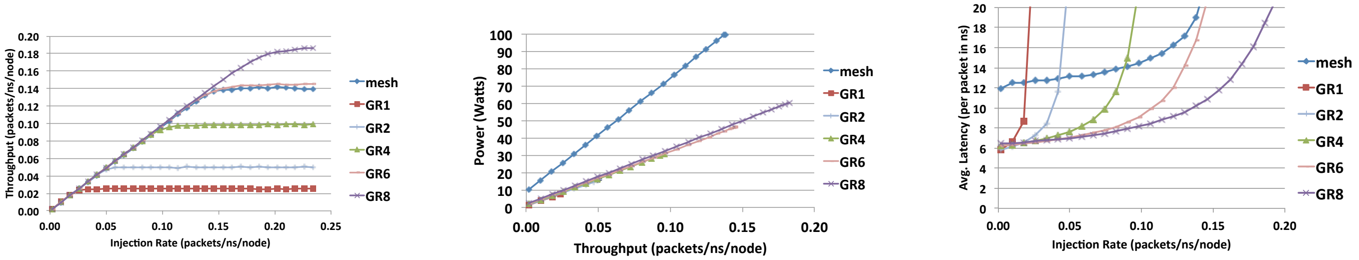

Figure 14: Energy proportionality of *Super-Star* topology with varying number of global routers:(a) Network Throughput (b) Network Power and (c) Network Latency.

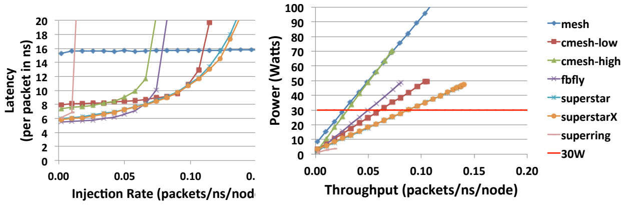

Figure 15: Network latency (a) and Network power (b) for equal wires study.

cmesh-high and *fbfly*, which benefits more from the additional bandwidth than the loss due to decreased router frequency. On the other hand, the narrower datapath of *Super-Star* and *Super-StarX* causes their throughput to saturate at a lower injection rate. Routers of *Super-Star* and *Super-StarX* become smaller due to the narrower channels, which results in better power efficiency, whereas the large channels of Mesh and *cmesh-high* increases switch power significantly. Similar to Figure 12(a), if we limit the network power to 30W across all topologies, the proposed *Super-Star* and *Super-StarX* topologies can provide 3× higher throughput than *mesh* and 1.4× higher throughput than *cmesh-low*.

5.3. Application Workloads

In this section, we study the characteristics of different topologies with real application workloads. We evaluate five multi-programmed workloads. The first four workloads, *Mix 1, Mix 2, Mix 3 and Mix 4*, run 46 copies of 12 unique applications which are chosen randomly from our suite of 35 single-threaded applications. The fifth workload, *SPLASH mix*, runs 64 threads each of 8 parallel applications taken from SPLASH-2 benchmark suite. The workloads are listed in Table 4 along with the total cache miss rate of each workload measured in terms of Misses Per Kilo Instructions (MPKI).

Figure 16(a) shows the system performance of various topologies and Figure 16(b) shows the network power consumption. We can observe that trends from our studies with statistical traffic persist. The *Super-StarX* topology provides an average performance improvement of 17% over the mesh topology while consuming 39% less power. Although the symmetric high-radix topologies and *Super-Ring* topology consume lower power, they have higher degradation in performance because they provide lower network throughput. The asymmetric high-radix topologies both improve performance and consumes lower power.

6. Related Work

In this paper, we study the scalability aspects of switch design and network topology design in the context of kilo-core processors. Below we summarize the closely related works.

6.1. Network-on-Chip Topologies

Today's multicore processors use a variety of interconnect topologies such as shared bus, rings, crossbars and meshes. The shared bus fabric was the prevalent interconnect design for decades because of low design complexity, low power consumption, and ability to support snoop-based coherence protocols. Unfortunately, buses do not scale beyond a few cores. Kumar et al. [25] showed that the shared bus fabric does not scale beyond 16 cores. To overcome scalability limitations, multicore processors adopted crossbars and rings.

The Niagara processor [24] implemented a crossbar interconnect to facilitate communication between 8 cores, 4 cache banks and I/O modules. Niagara's interconnect consisted of two 124b and 145b crossbars, operating at 1.2Ghz frequency in 65nm technology, providing a data bandwidth of 134.4GB/s, while consuming $\sim$ 3.8W of power. Recently, IBM Cyclops64 [46] supercomputer manycore processor chip implemented a 96-radix, 96b wide crossbar operating at 533MHz and occupying an area of 27mm^2.

Ring interconnects have been popular with multicore processors [7, 17, 40] due to relative simplicity of design of individual switches and the ability to provide global ordering. IBM Cell [7] has four 128b unidirectional rings operating at 1.6Ghz frequency and supporting data bandwidth of $\sim$ 200GB/s. ST's Spidergon [10] proposes a bidirectional ring augmented with links that directly connect nodes opposite to each other on the ring. These additional links reduce the average hop distance. To overcome bandwidth limitations, recent ring implementations use wide datapaths (e.g. Intel's Sandybridge [5] processors use 256b rings). Unfortunately, ring's bisection bandwidth does not scale with the number of nodes in the network, limiting it scalability to few dozens of cores.

The 2D mesh [6, 18, 35, 42, 45] topologies have become popular for tiled manycore processors because of their low complexity, planar 2D-layout properties and better scalability compared to rings. The TILE64 processor [45] implements five 32b 8x8 mesh networks to support various message classes and connect 64 nodes. Intel's Single Cloud Computing (SCC) [18] processor chip implements a 128b 6×4 concentrated mesh interconnect were two cores and two cache tiles share a router.

Table 4: List of workloads with their cache miss rates.

Workload	Applications	L1 MPKI	L2 MPKI
Mix 1	applu, astar, barnes, bzip2, calculix, gcc, gobmk, gromacs, hmmer, perlbench, sjeng, wrf	2,543	807
Mix 2	applu, bzip2, calculix, deal, gcc, gromacs, libquantum, perlbench, sap, sjeng, tonto, wrf	4,173	1,854
Mix 3	art, calculix, gobmk, gromacs, h264ref, libquantum, namd, ocean, omnet, perlbench, sap, sjas	7,211	3,119
Mix 4	astar, deal, Gems, gobmk, gromacs, lbm, leslie, mcf, milc, namd, sjeng, swim	15,899	9,263
SPLASH mix	Barnes, Cholesky, FFT, FMM, Lu, Ocean, Radix, Raytrace	4,096	1,408

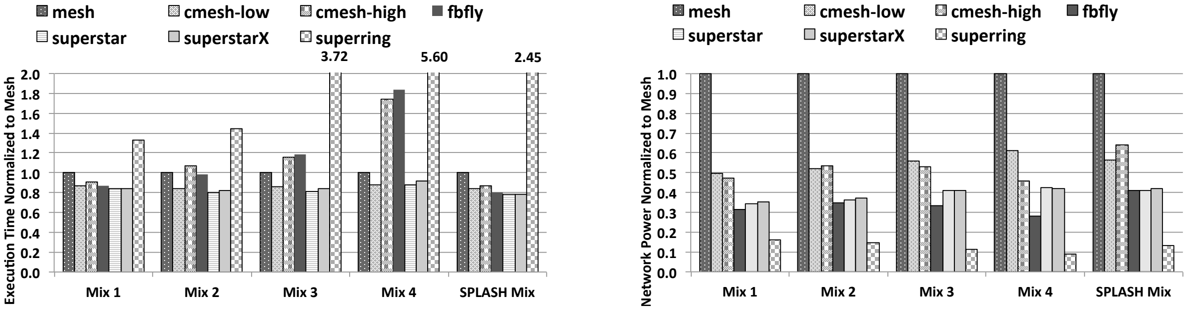

Figure 16: Performance of different topologies with application workloads: (a) Execution Time (a) and Network Power.

SCC's interconnect consumes $\sim 12W$ power while operating at $2Ghz$ frequency in 45nm technology.

Beyond commercial processor implementations, on-chip network topologies have been explored actively by academic researchers. Wang et al. [44] did a technology oriented, and power aware topology exploration for mesh/tori topologies with analytical models.

Several designs have been proposed to overcome the inefficiencies of 2D-meshes. Hierarchical bus-based topologies [13,43] have been proposed to reduce power consumption and minimize network latency. The bus-based proposals have limited scalability and were optimized for processors with 32 to 64 cores. Balfour and Dally proposed concentrated meshes [8] with express channels. Kim et al. [21] proposed flattened butterfly topology to reduce latency by providing rich connectivity.

Grot et al. [15] proposed multi-drop express channels (MECS) to reduce network latency by facilitating one-to-many communication over long express channels. The "multi-drop" concept of MECS topologies can be applied to the long channels in our proposed topologies to further improve network latency. However, the MECS topology can have significant buffering requirements to cover credit round trip delays over the express channels [16], as we scale up the network size. In [16], the authors discuss the challenges of scaling on-chip networks towards 100s of cores and propose use of the MECS topology to reduce cost of providing quality-of-service in network-on-chips with up to 256 nodes.

We believe, that while the above proposals were good designs which improved network latency significantly over the mesh topology, the design challenges and trade-offs for a kilo-core processor interconnect are different. Our proposed designs leverage the rich diversity of radix offered by *Swizzle-Switches* and our design space exploration is guided by *wire delay slack* leading to *asymmetric radix* designs. In our evaluations, we analyze the scalability of existing symmetric radix topologies such as concentrated meshes and flattened butterfly and compare our proposed designs to them.

Multi-stage fat trees [27], Reduced Unidirectional Fat Trees (RUFT) [34] and Clos [20] topologies have been also considered for on-chip networks. However, these proposals were based on traditional switch designs and thus limited all routers to radix-8. The Rigel 1000-core accelerator [19] proposes the use of a multi-stage tree interconnect.

In our design, routers with different radices operate at different frequencies. Prior work has exploited multiple frequency domains in 2D-mesh interconnects to manage congestion [29] and apply Dynamic Voltage Frequency Scaling (DVFS) [18].

6.2. High-Radix Switches

Prior works have recognized the multifaceted benefits of high-radix switches [21,22,23,39]. Kim et al. [23] proposed several optimizations to improve the scalability of switches with respect to radix. The optimizations included breaking down the arbitration into multiple local and global stages, decoupling the input and output virtual channel/switch allocation by including intermediate buffers at cross points and hierarchical crossbars with intermediate buffering.

Recently, Passas et al [30,31] proposed high-radix crossbar interconnects for 128 tile chips. Their implementation of a 128-radix crossbar was 32b wide, divided the data transfer into 3-stages and operated at a frequency of 750 Mhz at 90nm technology. The crossbar datapath occupied an area of 7.6mm^2, while the arbitration logic (or scheduler) is a iSLIP [28] scheduler and occupies an area of 7.2mm^2. Their work recognizes that arbitration complexity is a bottleneck in designing high-radix switches and proposes wiring optimizations to reduce the arbitration delay to 10ns.

In contrast to above decoupled approaches, *Swizzle-Switches* take an integrated approach towards arbitration to provide excellent scalability. The datapath and arbitration in a *Swizzle-Switch* is tightly coupled in a SRAM-like layout, reducing the area and critical path delay for the switch. Unlike traditional logic-tree arbiters, the arbitration in *Swizzle-Switch* is done by updating the internally stored priority bits on a cycle-by-cycle basis.

7. Conclusion

To realize kilo-core processors, it is important that we find a solution for designing a performance and power scalable on-chip interconnection network. In this paper, we proposed a class of asymmetric high-radix topologies that decouple local and global communication optimizations. Our proposed topologies employ the design principle that routers need to be only as fast as the wires that connect them. Thus, we employed fast medium-radix switches for local routers to achieve efficient local communication. Using a few high-radix global switches to connect local routers, we were able to reduce the hop count for global communication and also improve the overall throughput of the network.

Our experiments demonstrated that the best performing asymmetric high-radix topology improves average network latency over mesh by 45% while reducing the power consumption by 40%. When compared to symmetric high-radix topologies (i.e. concentrated meshes and flattened butterfly) our proposed topologies improve network throughput by 2.9× and network latency by 14% while providing similar power efficiency.

8. Acknowledgements

This research is supported by the National Science Foundation under the award CCF-1256203. We thank the generous support of our industrial sponsor, ARM Ltd.

References

[1] "AMD Opteron," http://www.amd.com/us/products/server/processors.

[2] "Azul Systems Vega 3," http://www.azulsystems.com/products/vega/processor.

[3] "Cavium Systems Octeon," http://www.cavium.com.

[4] "Intel Xeon E7," http://www.intel.com/content/www/us/en/processors/.

[5] "Intel's Sandy Bridge Architecture Exposed," http://www.anandtech.com/show/3922/intels-sandy-bridge-architecture-exposed/4.

[6] "Tilera TILE-Gx100," http://www.tilera.com/products/TILE-Gx.php.

[7] T. W. Ainsworth and T. M. Pinkston, "Characterizing the cell eib on-chip network," *IEEE Micro*, vol. 27, no. 5, pp. 6–14, 2007.

[8] J. Balfour and W. J. Dally, "Design tradeoffs for tiled cmp on-chip networks," in *ICS-20*, 2006.

[9] N. Binkert, B. Beckmann *et al.*, "The gem5 simulator," *Computer Architecture News (CAN)*, Jun 2011.

[10] L. Bononi and N. Concer, "Simulation and analysis of network-on-chip architectures: Ring, spidergon and 2d mesh," in *DATE*, 2006.

[11] S. Borkar, "Networks for multi-core chips: A contrarian view," *Special Session at ISLPED*, 2007.

[12] ——, "Thousand core chips: a technology perspective," in *DAC-44*, 2007.

[13] R. Das, S. Eachempati, A. K. Mishra, N. Vijaykrishnan, and C. R. Das, "Design and evaluation of a hierarchical on-chip interconnect for next-generation cmps," in *HPCA-15*, 2009.

[14] R. Dreslinski, K. Sewell, T. Manville, S. Satpathy, N. Pinckney, G. Blake, M. Cieslak, R. Das, T. Wenisch, D. Sylvester, D. Blaauw, and T. Mudge, "Swizzle-switch: A self-arbitrating high-radix crossbar for noc systems," in *HotChips*, 2012.

[15] B. Grot, J. Hestness, S. W. Keckler, and O. Mutlu, "Express cube topologies for on-chip interconnects," in *HPCA-15*, 2009.

[16] ——, "Kilo-noc: a heterogeneous network-on-chip architecture for scalability and service guarantees," in *ISCA-38*, 2011.

[17] M. Gschwind, H. Hofstee *et al.*, "Synergistic processing in cell's multicore architecture," *IEEE Micro*, vol. 26, no. 2, pp. 10–24, 2006.

[18] J. Howard, S. Dighe, Y. Hoskote, S. R. Vangal, D. Finan, G. Ruhl, D. Jenkins, H. Wilson, N. Borkar, G. Schrom, F. Pailet, S. Jain, T. Jacob, S. Yada, S. Marella, P. Salihundam, V. Erraguntla, M. Konow, M. Riepen, G. Droege, J. Lindemann, M. Gries, T. Apel, K. Henriss, T. Lund-Larsen, S. Steibl, S. Borkar, V. De, R. F. V. der Wijngaart, and T. G. Mattson, "A 48-core ia-32 message-passing processor with dvfs in 45nm cmos," in *ISSCC*, 2010.

[19] D. R. Johnson, M. R. Johnson, J. H. Kelm, W. Tuohy, S. S. Lumetta, and S. J. Patel, "Rigel: A 1,024-core single-chip accelerator architecture," *IEEE Micro*, vol. 31, no. 4, pp. 30–41, 2011.

[20] Y.-H. Kao, N. Alfaraj, M. Yang, and H. J. Chao, "Design of high-radix clos network-on-chip," in *NOCS*, 2010.

[21] J. Kim, J. Balfour, and W. Dally, "Flattened butterfly topology for on-chip networks," in *MICRO-40*, 2007.

[22] J. Kim, W. J. Dally, and D. Abts, "Flattened butterfly: a cost-efficient topology for high-radix networks," in *ISCA-34*, 2007.

[23] J. Kim, W. J. Dally, B. Towles, and A. K. Gupta, "Microarchitecture of a high-radix router," in *ISCA-32*, 2005.

[24] P. Kongetira, K. Aingaran, and K. Olukotun, "Niagara: A 32-way multithreaded sparc processor," *IEEE Micro*, vol. 25, no. 2, pp. 21–29, 2005.

[25] R. Kumar, V. Zyuban, and D. M. Tullsen, "Interconnections in multicore architectures: Understanding mechanisms, overheads and scaling," in *ISCA-32*, 2005.

[26] J. W. Lee, M. C. Ng, and K. Asanovic, "Globally-Synchronized Frames for Guaranteed Quality-of-Service in On-Chip Networks," in *ISCA-35*, 2008.

[27] D. Ludovici, F. G. Villamón, S. Medardoni, C. G. Requena, M. E. Gómez, P. López, G. N. Gaydadjiev, and D. Bertozzi, "Assessing fat-tree topologies for regular network-on-chip design under nanoscale technology constraints," in *DATE*, 2009.

[28] N. McKeown, "The islip scheduling algorithm for input-queued switches," *Networking, IEEE/ACM Transactions on*, 1999.

[29] A. K. Mishra, R. Das, S. Eachempati, V. Narayanan, R. Iyer, and C. R. Das, "A Case for Dynamic Frequency Tuning in On-Chip Networks," in *MICRO-42*, 2009.

[30] G. Passas, M. Katevenis, and D. Pnevmatikatos, "Vlsi micro-architectures for high-radix crossbar schedulers," in *NOCS*, 2011.

[31] ——, "A 128 x 128 x 24gb/s crossbar interconnecting 128 tiles in a single hop and occupying 6% of their area," in *NOCS*, 2010.

[32] H. Patil, R. Cohn, M. Charney, R. Kapoor, A. Sun, and A. Karunanidhi, "Pinpointing Representative Portions of Large Intel Itanium Programs with Dynamic Instrumentation," in *MICRO-37*, 2004.

[33] L.-S. Peh and W. J. Dally, "A Delay Model and Speculative Architecture for Pipelined Routers," in *HPCA-8*, 2001.

[34] C. G. Requena, F. G. Villamón, M. E. Gómez, P. J. L. Rodríguez, and J. Duato, "Ruft: Simplifying the fat-tree topology," in *ICPADS*, 2008.

[35] K. Sankaralingam, R. Nagarajan, H. Liu, C. Kim, J. Huh, D. Burger, S. W. Keckler, and C. R. Moore, "Exploiting ILP, TLP, and DLP with The Polymorphous TRIPS Architecture," in *ISCA-30*, 2003.

[36] S. Satpathy, R. Das, R. Dreslinski, D. Sylvester, T. Mudge, and D. Blaauw, "High radix self-arbitrating switch fabric with multiple arbitration schemes and quality of service," in *DAC-49*, 2012.

[37] S. Satpathy, R. Dreslinski, T.-C. Ou, D. Sylvester, T. Mudge, and D. Blaauw, "Swift: A 2.1tb/s 32x32 self-arbitrating manycore interconnect fabric," in *VLSIC*, 2011.

[38] S. Satpathy, K. Sewell, T. Manville, Y.-P. Chen, R. G. Dreslinski, D. Sylvester, T. N. Mudge, and D. Blaauw, "A 4.5tb/s 3.4tb/s/w 64x64 switch fabric with self-updating least recently granted priority and quality of service arbitration in 45nm cmos," in *ISSCC*, 2012.

[39] S. Scott, D. Abts, J. Kim, and W. J. Dally, "The blackwidow high-radix clos network," in *ISCA-33*, 2006.

[40] L. Seiler and D. e. a. Carmean, "Larrabee: a many-core x86 architecture for visual computing," *ACM Transactions on Graphics*, 2008.

[41] K. Sewell, R. Dreslinski, T. Manville, S. Satpathy, N. Pinckney, G. Blake, M. Cieslak, R. Das, T. Wenisch, D. Sylvester, D. Blaauw, and T. Mudge, "Swizzle-switch networks for many-core systems," in *JETCAS*, 2012.

[42] M. B. Taylor, J. S. Kim, J. E. Miller, D. Wentzlaff, F. Ghodrat, B. Greenwald, H. Hoffmann, P. Johnson, J.-W. Lee, W. Lee, A. Ma, A. Saraf, M. Seneski, N. Shnidman, V. Strumpen, M. Frank, S. P. Amarasinghe, and A. Agarwal, "The raw microprocessor: A computational fabric for software circuits and general-purpose programs," *IEEE Micro*, vol. 22, no. 2, pp. 25–35, 2002.

[43] A. N. Udipi, N. Muralimanohar, and R. Balasubramonian, "Towards scalable, energy-efficient, bus-based on-chip networks," in *HPCA-16*, 2010.

[44] H. Wang, L.-S. Peh, and S. Malik, "A technology-aware and energy-oriented topology exploration for on-chip networks," in *DATE*, 2005.

[45] D. Wentzlaff, P. Griffin *et al.*, "On-chip interconnection architecture of the tile processor," *IEEE Micro*, vol. 27, no. 5, pp. 15–31, 2007.

[46] Y. P. Zhang *et al.*, "A study of the on-chip interconnection network for the ibm cyclops64 multi-core architecture," in *IPDPS-20*, 2006.