Kilo-NOC: A Heterogeneous Network-on-Chip Architecture for Scalability and Service Guarantees

Boris Grot[1] bgrot@cs.utexas.edu
Joel Hestness[1] hestness@cs.utexas.edu
Stephen W. Keckler[1,2] skeckler@nvidia.com
Onur Mutlu[3] onur@cmu.edu

[1]The University of Texas at Austin
Austin, TX

[2]NVIDIA
Santa Clara, CA

[3]Carnegie Mellon University
Pittsburgh, PA

ABSTRACT

Today's chip-level multiprocessors (CMPs) feature up to a hundred discrete cores, and with increasing levels of integration, CMPs with hundreds of cores, cache tiles, and specialized accelerators are anticipated in the near future. In this paper, we propose and evaluate technologies to enable networks-on-chip (NOCs) to support a thousand connected components (Kilo-NOC) with high area and energy efficiency, good performance, and strong quality-of-service (QOS) guarantees. Our analysis shows that QOS support burdens the network with high area and energy costs. In response, we propose a new lightweight topology-aware QOS architecture that provides service guarantees for applications such as consolidated servers on CMPs and real-time SOCs. Unlike prior NOC quality-of-service proposals which require QOS support at every network node, our scheme restricts the extent of hardware support to portions of the die, reducing router complexity in the rest of the chip. We further improve network area- and energy-efficiency through a novel flow control mechanism that enables a single-network, low-cost elastic buffer implementation. Together, these techniques yield a heterogeneous Kilo-NOC architecture that consumes 45% less area and 29% less power than a state-of-the-art QOS-enabled NOC without these features.

Categories and Subject Descriptors:
C.1.4 [Computer Systems Organization]: Multiprocessors – *Interconnection architectures*

General Terms: Design, Measurement, Performance

1. INTRODUCTION

Complexities of scaling single-threaded performance have pushed processor designers in the direction of chip-level integration of multiple cores. Today's state-of-the-art general-purpose chips integrate up to one hundred cores [27, 28], while GPUs and other specialized processors may contain hundreds of execution units [24]. In addition to the main processors, these chips often integrate cache memories, specialized accelerators, memory controllers, and other resources. Likewise, modern systems-on-a-chip (SOCs) contain many cores, accelerators, memory channels, and interfaces. As the degree of integration increases with each technology generation, chips containing over a thousand discrete execution and storage resources will be likely in the near future.

Chip-level multiprocessors (CMPs) require an efficient communication infrastructure for operand, memory, coherence, and control transport [29, 8, 31], motivating researchers to propose structured on-chip networks as replacements to buses and ad-hoc wiring solutions of single-core chips [5]. The design of these networks-on-chip (NOCs) typically requires satisfaction of multiple conflicting constraints, including minimizing packet latency, reducing router area, and lowering communication energy overhead. In addition to basic packet transport, future NOCs will be expected to provide certain advanced services. In particular, *quality-of-service (QOS)* is emerging as a desirable feature due to the growing popularity of server consolidation, cloud computing, and real-time demands of SOCs. Despite recent advances aimed at improving the efficiency of individual NOC components such as buffers, crossbars, and flow control mechanisms [22, 30, 15, 18], as well as features such as QOS [19, 10], little attention has been paid to network scalability beyond several dozen terminals.

In this work, we focus on NOC scalability from the perspective of energy, area, performance, and quality-of-service. With respect to QOS, our interest is in mechanisms that provide hard guarantees, useful for enforcing Service Level Agreement (SLA) requirements in the cloud or real-time constraints in SOCs. Prior work showed that a direct low-diameter topology improves latency and energy efficiency in NOCs with dozens of nodes [16, 9]. While our analysis confirms this result, we identify critical scalability bottlenecks in these topologies once scaled to configurations with hundreds of network nodes. Chief among these is the buffer overhead associated with large credit round-trip times of long channels. Large buffers adversely affect NOC area and energy efficiency. The addition of QOS support further increases storage overhead, virtual channel (VC) requirements, and arbitration complexity. For instance, a 256-node NOC with a low-diameter Multidrop Express Channel (MECS) topology [9] and Preemptive Virtual Clock (PVC) QOS mechanism [10] may require 750 VCs per router and over 12 MBs of buffering per chip, as shown in Sec. 3.1.

[‡] ©ACM, (2011). This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version appears in the Proceedings of ISCA 2011.

ISCA'11, June 4–8, 2011, San Jose, California, USA.
Copyright 2011 ACM 978-1-4503-0472-6/11/06 ...$10.00.

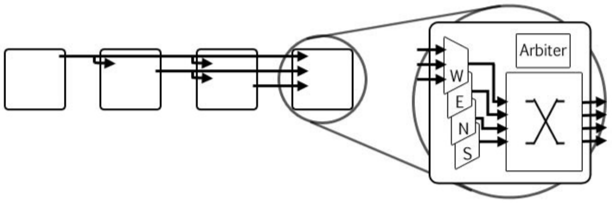


Figure 1: Multidrop Express Channel architecture.

In this paper, we propose a hybrid NOC architecture that offers low latency, small footprint, good energy efficiency, and SLA-strength QOS guarantees. The architecture is designed to scale to a large number of on-chip nodes and is evaluated in the context of a thousand terminal (Kilo-NOC) system. To reduce the substantial QOS-related overheads, we address a key limitation of prior NOC QOS approaches which have required hardware support at every router node. Instead, our proposed topology-aware QOS architecture consolidates shared resources (e.g. memory controllers) within a portion of the network and only enforces QOS within subnetworks that contain these shared resources. The rest of the network, freed from the burden of hardware QOS support, enjoys diminished cost and complexity. Our approach relies on a richly-connected low-diameter topology to enable single-hop access to any QOS-protected subnetwork, effectively eliminating intermediate nodes as sources of interference. To our knowledge, this work is the first to consider the interaction between topology and quality-of-service.

Despite a significant reduction in QOS-related overheads, buffering remains an important contributor to our router area and energy footprint. We eliminate much of the expense by introducing a light-weight *elastic buffer (EB)* architecture that integrates storage directly into links, again using the topology to our advantage. To avoid deadlock in the resulting network, our approach leverages the multi-drop capability of a MECS interconnect to establish a dynamically allocated escape path for blocked packets into intermediate routers along the channel. In contrast, earlier EB schemes required multiple networks or many virtual channels for deadlock-free operation, incurring significant area and wire cost [21]. In a kilo-terminal network, the proposed single-network elastic buffer architecture requires only two virtual channels and reduces router storage requirements by 8x over a baseline MECS router without QOS support and by 12x compared to a QOS-enabled design.

Our results show that these techniques synergistically work to improve performance, area, and energy efficiency. In a kilo-terminal network in 15 nm technology, our final QOS-enabled NOC design reduces network area by 30% versus a modestly-provisioned MECS network with no QOS support and 45% compared to a MECS network with PVC, a prior NOC QOS architecture. Network energy efficiency improved by 29% and 40% over MECS without and with QOS support, respectively, on traffic with good locality. On random traffic, the energy savings diminish to 20% and 29% over the respective MECS baselines as wire energy dominates router energy consumption. Our NOC obtains both area and energy benefits without compromising either performance or QOS guarantees. In a notional $256mm^2$ high-end chip, the proposed NOC consumes under 7% of the overall area and 23.5W of power at a sustained network load of 10%, a modest fraction of the overall power budget.

Table 1: Scalability of NOC topologies. k: network radix, v: per-port VC count, C: a small integer.

	Mesh	FBfly	MECS
Network diameter	$2 \cdot k$	2	2
Bisection channels/dimension	2	$k^2/2$	k
Buffers	C	k^2	k^2
Crossbar (network ports)	4×4	$k \times k$	4×4
Arbitration	$log(4v)$	$log(k \cdot v)$	$log(k \cdot v)$

2. BACKGROUND

This section reviews key NOC concepts, draws on prior work to identify important Kilo-NOC technologies, and analyzes their scalability bottlenecks. We start with conventional NOC attributes – topology, flow control, and routing – followed by quality-of-service technologies.

2.1 Conventional NOC Attributes

2.1.1 Topology

Network topology determines the connectivity among nodes and is therefore a first-order determinant of network performance and energy-efficiency. To avoid the large hop counts associated with rings and meshes of early NOC designs [25, 29], researchers have turned to richly-connected low-diameter networks that leverage the extensive on-chip wire budget. Such topologies reduce the number of costly router traversals at intermediate hops, thereby improving network latency and energy efficiency, and constitute a foundation for a Kilo-NOC.

One low-diameter NOC topology is the flattened butterfly (FBfly), which maps a richly-connected butterfly network to planar substrates by fully interconnecting nodes in each of the two dimensions via dedicated point-to-point channels [16]. An alternative topology called Multidrop Express Channels (MECS) uses point-to-multipoint channels to also provide full intra-dimension connectivity but with fewer links [9]. Each node in a MECS network has four output channels, one per cardinal direction. Light-weight drop interfaces allow packets to exit the channel into one of the routers spanned by the link. Figure 1 shows the high-level architecture of a MECS channel and router.

Scalability: Potential scalability bottlenecks in low-diameter networks are channels, input buffers, crossbar switches, and arbiters. The scaling trends for these structures are summarized in Table 1. The flattened butterfly requires O(k^2) bisection channels per row/column, where k is the network radix, to support all-to-all intra-dimension connectivity. In contrast, the bisection channel count in MECS grows linearly with the radix.

Buffer capacities need to grow with network radix, assumed to scale with technology, to cover the round-trip credit latencies of long channel spans. Doubling the network radix doubles the number of input channels *and* the average buffer depth at an input port, yielding a *quadratic* increase in buffer capacity per node. This relationship holds for both flattened butterfly and MECS topologies and represents a true scalability obstacle.

Crossbar complexity is also quadratic in the number of input and output ports. This feature is problematic in a flattened butterfly network, where port count grows in proportion to the network radix and causes a quadratic increase in switch area for every 2x increase in radix. In a MECS network, crossbar area stays nearly constant as the number of output ports is fixed at four and each switch input port is multiplexed among all network inputs from the same direction (see Figure 1). While switch complexity is not a concern in MECS, throughput can suffer because of the asymmetry in the number of input and output ports.

Finally, arbitration complexity grows logarithmically with port count. Designing a single-cycle arbiter for a high-radix router with a fast clock may be a challenge; however, arbitration can be pipelined over multiple cycles. While pipelined arbitration increases node delay, it is compensated for by the small hop count of low-diameter topologies. Hence, we do not consider arbitration a scalability bottleneck.

2.1.2 Flow Control

Flow control governs the flow of packets through the network by allocating channel bandwidth and buffer slots to packets. Conventional interconnects have traditionally employed packet-granularity bandwidth and storage allocation, exemplified by Virtual Cut-Through (VCT) flow control [14]. In contrast, NOCs have relied on flit-level flow control [4], refining the allocation granularity to reduce the per-node storage requirements.

Scalability: In a Kilo-NOC with a low-diameter topology, long channel traversal times necessitate deep buffers to cover the round-trip credit latency. At the same time, wide channels reduce the number of flits per network packet. These two trends diminish the benefits of flit-level allocation since routers typically have enough buffer capacity for multiple packets. In contrast, packet-level flow control couples bandwidth and storage allocation, reducing the number of required arbiters, and amortizes the allocation delay over the length of a packet. Thus, in a Kilo-NOC, packet-level flow control is preferred to a flit-level architecture.

Elastic buffering: Recent research has explored the benefits of integrating storage elements, referred to as *elastic buffers (EB)*, directly into network links. Kodi et al. proposed a scheme called *iDEAL* that augments a conventional virtual-channel architecture with in-link storage, demonstrating savings in buffer area and power [17]. An alternative proposal by Michelogiannakis et al. advocates a pure elastic-buffered architecture without any virtual channels [21]. To prevent protocol deadlock in the resulting wormhole-routed NOC, the scheme requires a dedicated network for each packet class.

Scalability: To prevent protocol deadlock due to the serializing nature of buffered links, iDEAL must reserve a virtual channel at the destination router for each packet. As a result, its router buffer requirements in a low-diameter NOC grow quadratically with network radix as explained in Section 2.1.1, impeding scalability. A pure elastic-buffered architecture enjoys linear scaling in router storage requirements, but needs multiple networks for deadlock avoidance, incurring chip area and wiring expense.

2.1.3 Routing

A routing function determines the path of a packet from its source to the destination. Most networks use deterministic routing schemes, whose chief appeal is simplicity. In contrast, adaptive routing can boost throughput of a given topology at the cost of additional storage and/or allocation complexity.

Scalability: The scalability of a routing algorithm is a function of the path diversity attainable for a given set of channel resources. Compared to rings and meshes, direct low-diameter topologies typically offer greater path diversity through richer channel resources. Adaptive routing on such topologies has been shown to boost throughput [16, 9]; however, the gains come at the expense of energy efficiency due to the overhead of additional router traversals. While we do not consider routing a scalability bottleneck, reliability requirements may require additional complexity not considered in this work.

2.2 Quality-of-Service

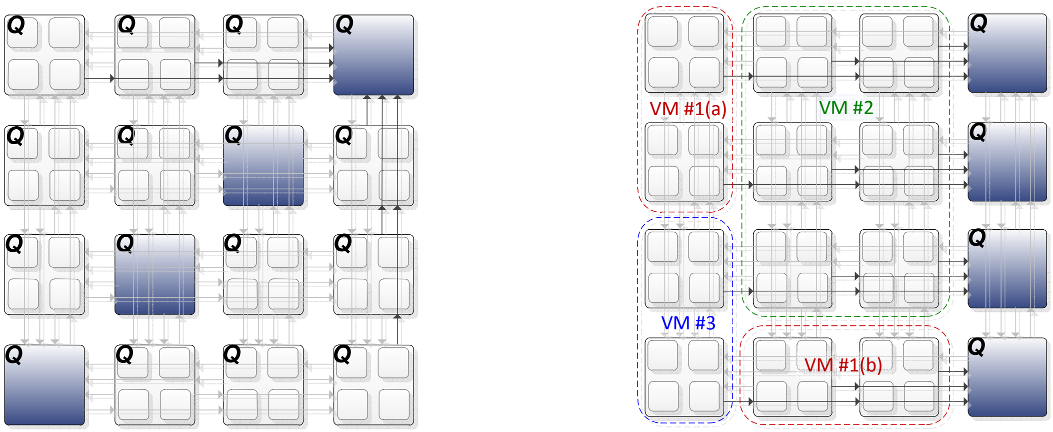
Cloud computing, server consolidation, and real-time applications demand on-chip QOS support for security, performance isolation, and guarantees. In many cases, a software layer will be unable to meet QOS requirements due to the fine-grained nature of chip-level resource sharing. Thus, we anticipate that hardware quality-of-service infrastructure will be a desirable feature in future CMPs. Unfortunately, existing network QOS schemes represent a weighty proposition that conflicts with the objectives of an area- and energy-scalable NOC.

Current network QOS schemes require dedicated per-flow packet buffers at all network routers or source nodes [7, 19], resulting in costly area and energy overheads. Recently proposed Preemptive Virtual Clock (PVC) architecture for NOC QOS relaxes the buffer requirements by using *preemption* to guarantee freedom from priority inversion [10]. Under PVC, routers are provisioned with a minimum number of virtual channels (VCs) to cover the round-trip credit delay of a link. Without dedicated buffer resources for each flow, lower priority packets may block packets with higher dynamic priority. PVC detects such priority inversion situations and resolves them through preemption of lower-priority packets. Discarded packets require retransmission, signaled via a dedicated ACK network.

Scalability: While PVC significantly reduces QOS cost over prior work, in a low-diameter topology its VC requirements grow quadratically with network radix (analysis is similar to the one in Section 2.1.1), impeding scalability. VC requirements grow because multiple packets are not allowed to share a VC to prevent priority inversion within a FIFO buffer. Thus, longer links require more, but not deeper, VCs. Large VC populations adversely affect both storage requirements and arbitration complexity. In addition, PVC maintains per-flow state at each router whose storage requirements grow linearly with network size. Finally, preemption events in PVC incur energy and latency overheads proportional to network diameter and preemption frequency. These considerations argue for an alternative network organization that provides QOS guarantees without compromising efficiency.

2.3 Summary

Kilo-scale NOCs require low-diameter topologies, aided by efficient flow control and routing mechanisms, to minimize energy and delay overheads of multi-hop transfers. While researchers have proposed low-diameter topologies for on-chip interconnects, their scalability with respect to area, energy, and performance has not been studied. Our analysis shows that channel requirements and switch complexity are not true scalability bottlenecks, at least for some topology choices. On the other hand, buffer demands scale quadrat-



(a) Baseline QOS-enabled CMP

(b) Topology-aware QOS approach

Figure 2: 64-tile CMP with 4-way concentration and MECS topology. Light nodes: core+cache tiles; shaded nodes: memory controllers; Q: QOS hardware. Dotted lines: *domains* in a topology-aware QOS architecture.

ically with network radix, diminishing area- and energy-efficiency of large-scale low-diameter NOCs. Quality-of-service further increases storage demands and creates additional overheads. Supporting tomorrow's Kilo-NOC configurations requires addressing these scalability bottlenecks.

3. KILO-NOC ARCHITECTURE

3.1 Baseline Design

Our target in this work is a 1024-tile CMP in 15 nm technology. Figure 2(a) shows the baseline organization, scaled down to 64 tiles for clarity. Light nodes in the figure integrate core and cache tiles; shaded nodes represent shared resources, such as memory controllers; 'Q' indicates hardware QOS support at the node. We employ concentration [1] to reduce the number of network nodes to 256 by integrating four terminals at a single router via a fast crossbar switch. A *node* refers to a network node, while a *terminal* is a discrete system resource, such as a core, cache tile, or memory controller, with a dedicated port at a network node. The nodes are interconnected via a richly connected MECS topology. We choose MECS due to its low diameter, scalable channel count, modest switch complexity, and unique capabilities offered by multidrop. QOS guarantees are enforced by PVC.

The 256 concentrated nodes in our kilo-terminal network are arranged in a 16 by 16 grid. Each MECS router integrates 30 network input ports (15 per dimension). With one cycle of wire latency between adjacent nodes, maximum channel delay, from one edge of the chip to another, is 15 cycles. The following equation gives the maximum round-trip credit time, t_{RTCT} [6]:

$$t_{RTCT} = 2t_{wire} + t_{flit} + t_{credit} + 1 \qquad (1)$$

where t_{wire} is the one-way wire delay, t_{flit} is the flit pipeline latency, and t_{credit} is the credit pipeline latency. With a three stage router datapath and one cycle for credit processing, the maximum t_{RTCT} in the above network is 35 cycles. This represents a lower bound for per-port buffer requirements in the absence of any location-dependent optimizations. Dedicated buffering for each packet class, necessary for deadlock avoidance, and QOS demands impose additional overheads.

In the case of QOS, packets from different flows generally require separate virtual channels to prevent priority inversion within a single VC FIFO. To accommodate a worst-case pattern consisting of single-flit packets from different flows, an unoptimized router would require 35 VCs per port. Several optimizations could be used to reduce the VC and buffer requirements at additional design expense and arbitration complexity. As the potential optimization space is large, we simply assume that a 25% reduction in per-port VC requirements can be achieved. To accommodate a maximum packet size of four flits, a baseline QOS router features 25 four-deep VC's per port for a total population of 750 VCs and 3000 flit slots per 30-port router. With 16-byte flits, total storage required is 48 KB per router and 12 MB network-wide.

Without QOS support, each port requires just one VC per packet class. With two priority levels (Request at low priority and Reply at high priority), a pair of 35-deep virtual channels is sufficient for deadlock avoidance while covering the maximum round-trip credit delay. The required per-port buffering is thus 70 flits compared to 100 flits in a QOS-enabled router (25 VCs with 4 flits per VC).

3.2 Topology-aware QOS Architecture

Our first optimization target is the QOS mechanism. As noted in Section 2.2, QOS imposes a substantial virtual channel overhead in a low-diameter topology, aggravating storage requirements and arbitration complexity. In this work, we take a topology-aware approach to on-chip quality-of-service. While existing network quality-of-service architectures demand dedicated QOS logic and storage at every router, we seek to limit the number of nodes requiring hardware QOS support. Our proposed scheme isolates shared resources into one or more dedicated regions of the network, called *shared regions (SRs)*, with hardware QOS enforcement within each SR. The rest of the network is freed from the burden of hardware QOS support and enjoys reduced cost and complexity.

The Topology-Aware QOS (TAQ) architecture leverages the rich intra-dimension connectivity afforded by MECS (or another low-diameter topology) to ensure single-hop access to any shared region, which we achieve by organizing the SRs into columns spanning the entire width of the die. Single-hop connectivity guarantees interference-free transit into an SR. Once inside the shared region, a packet is regulated by the deployed QOS mechanism as it proceeds to its destination, such as a memory controller. To prevent unregulated contention for network bandwidth at concentrated nodes outside of the SR, we require the OS or hypervisor to co-schedule only threads from the same virtual machine onto a node*. Figure 2(b) shows the proposed organization. While in the figure the SR column is on the edge of the die, such placement is not required by TAQ.

Threads running under the same virtual machine on a CMP benefit from efficient support for on-chip data sharing. We seek to facilitate both intra-VM and inter-VM data sharing while preserving performance isolation and guarantees. We define the *domain* of a VM to be the set of nodes allocated to it. The objective is to provide service guarantees for each domain across the chip. The constraint is that QOS is explicitly enforced only inside the shared regions. We achieve the desired objective via the following rules governing the flow of traffic:

1. Communication within a dimension is unrestricted, as the MECS topology provides interference-free single-hop communication in a given row or column.
2. Dimension changes are unrestricted *iff* the turn node belongs to the same domain as the packet's source or destination. For example, all cache-to-cache traffic associated with VM #2 in Figure 2(b) stays within a single convex region and never needs to transit through a router in another domain.
3. Packets requiring a dimension change at a router from an unrelated domain must flow through one of the shared regions. Depending on the locations of the communicating nodes with respect to the SRs, the resulting routes may be non-minimal. For instance, in Figure 2(b), traffic from partition (a) of VM #1 transiting to partition (b) of the same VM must take the longer path through the shared column to avoid turning at a router associated with VM #2. Similarly, traffic between different VMs, such as inter-VM shared page data, may also need to flow through a shared region.

Our proposal preserves guarantees for all flows regardless of the locations of communicating nodes. Nonetheless, performance and energy-efficiency can be maximized by reducing a VM's network diameter. Particularly effective are placements that form convex-shaped domains, as they localize traffic and improve communication efficiency. Recent work by Marty and Hill examining cache coherence policies in the context of consolidated servers on a CMP reached similar conclusions regarding benefits of VM localization [20].

Summarizing, our QOS architecture consists of three components: a richly-connected topology, QOS-enabled shared regions, and OS/hypervisor scheduling support.

Topology: TAQ requires a topology with a high degree of connectivity to physically isolate traffic between non-adjacent routers. While this work uses MECS, other topologies, such as a flattened butterfly are possible as well. We exploit the connectivity to limit the extent of hardware QOS support to a few confined regions of the chip, which can be reached in one hop from any node. With XY dimension-ordered routing (DOR), the shared resource regions must be organized as columns on the two-dimensional grid of nodes to maintain the single-hop reachability property.

Shared regions: TAQ concentrates resources that are shared across domains, such as memory controllers or accelerators, into dedicated, QOS-enabled regions of the die. In this work, we assume that cache capacity is shared within a domain but not across domains, which allows us to elide QOS support for caches. If necessary, TAQ can easily be extended to include caches.

The shared resource regions serve two purposes. The first is to ensure fair or differentiated access to shared resources. The second is to support intra- and inter-VM communication for traffic patterns that would otherwise require a dimension change at a router from an unrelated domain.

Scheduling support: We rely on the operating system to 1) control thread placement at concentrated nodes outside of the SR, and 2) assign bandwidth or priorities to flows, defined at the granularity of a thread, application, or virtual machine, by programming memory-mapped registers at QOS-enabled routers. As existing OS/hypervisors already provide scheduling services and support different process priorities, the required additions are small.

3.3 Low-Cost Elastic Buffering

Freed from the burden of enforcing QOS, routers outside of the shared regions can enjoy a significant reduction in the number of virtual channels to just one VC per packet class. As noted in Sec. 3.1, a MECS router supporting two packet priority classes and no QOS hardware requires 30% fewer flit buffers than a QOS-enabled design. To further reduce storage overheads, we propose integrating storage into links by using a form of elastic buffering. Normally, elastic buffered networks are incompatible with QOS due to the serializing nature of EB flow control, which can introduce priority inversion within a channel. However, the proposed topology-aware QOS architecture enables elastic buffering outside of the shared regions by eliminating interference among flows from different VMs. Inside SRs, conventional buffering and flow control are still needed for traffic isolation and prioritization.

Point-to-point EB networks investigated in prior work do not reduce the minimum per-link buffer requirements, as storage in such networks is simply shifted from routers to links. We make the observation that in a point-to-multipoint MECS topology, elastic buffering can actually decrease overall storage requirements since each buffer slot in a channel is effectively shared by all downstream destination nodes. Thus, an EB-enhanced MECS network can be effective in diminishing buffer area and power. Unfortunately, existing EB architectures require significant virtual channel resources or multiple networks for avoiding protocol deadlock, as noted in Section 2.1.2. The resulting area and wire overheads diminish the appeal of elastic buffering.

3.3.1 Proposed EB Architecture

In this work, we propose an elastic buffer organization that affords considerable area savings over earlier schemes. Our approach combines elastic-buffered links with minimal virtual channel resources, enabling a single-network archi-

*Without loss of generality, we assume that QOS is used to provide isolation among VMs. Our approach can easily be adapted for application-level quality-of-service.

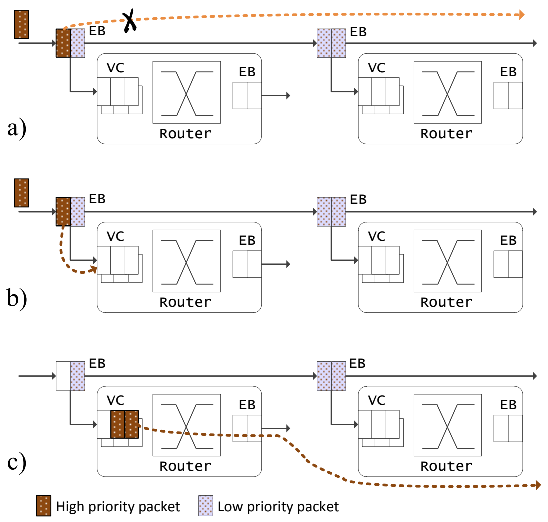


Figure 3: Elastic buffer deadlock avoidance.

tecture with hybrid EB/VC flow control. Unlike the iDEAL scheme, which also uses a hybrid organization, our architecture does not reserve a virtual channel for a packet at the sending router. Instead, a VC is allocated on-the-fly directly from an elastic buffer in the channel. Since neither buffer nor virtual channel resources are reserved upstream, VC requirements are not dependent on the link flight time. This approach provides a scalable alternative to iDEAL, whose VC requirements are proportional to the link delay and result in high buffer costs in future low-diameter NOCs.

Without pre-allocated buffer space at the target node, a network with elastic-buffered channels is susceptible to protocol deadlock. Deadlock can arise because low priority packets in the channel may prevent higher priority packets from reaching their destinations. To overcome potential deadlock, we exploit the multi-drop aspect of MECS channels to establish a dynamically allocated escape path into an intermediate router along a packet's direction of travel. We introduce a new flow control mechanism called Just-in-Time VC binding (JIT-VC), which enables packets in the channel to acquire a VC from an elastic buffer. Under normal operation, a packet will allocate a VC once it reaches the elastic buffer at the target (turn or destination) node. However, should a high priority (e.g., reply) packet be blocked in the channel, it can leverage the multi-drop capability of MECS to escape into an intermediate router via a JIT-allocated VC. Once buffered at an escape router, a packet will switch to a new MECS channel by traversing the router pipeline like any other packet. To prevent circular deadlock, we do not allow packets to switch dimensions at an escape node.

Figure 3 shows a high-level depiction of our approach. In (a), a high-priority packet in a MECS channel is obstructed by a low-priority one; (b) shows the blocked packet dynamically acquiring a buffer at a router associated with the EB; in (c), the high-priority packet switches to a new MECS channel and proceeds toward its destination.

The rerouting feature of the proposed deadlock avoidance scheme allows for packets at the same priority level to be reordered. If the semantics of the system require a predictable message order, than ordering may need to be enforced at the end points.

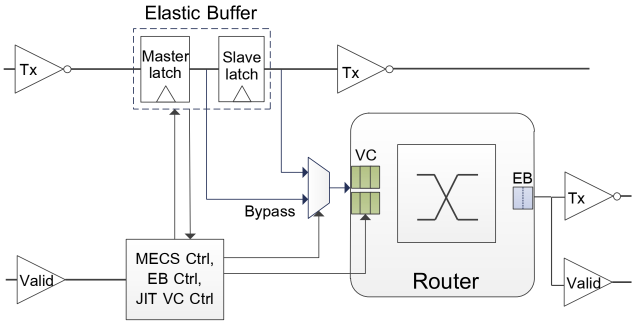


Figure 4: MECS with deadlock-free elastic buffer.

Figure 4 shows the proposed design in the context of a MECS network. The EB, based on the design by Michelogiannakis et al. [21], uses a master-slave latch combination that can store up to two flits. We integrate an EB into each drop interface along a MECS channel and augment the baseline elastic buffer with a path from the master latch to the router input port. A path from the slave latch to the router already exists for normal MECS operation, necessitating a mux to select between the two latches. We also add logic into the EB control block to query and allocate router-side VCs. This setup allows high priority packets to reactively escape blocked channels by dynamically allocating a VC, draining into a router, and switching to another MECS link.

3.3.2 *Deadlock Freedom*

We achieve deadlock freedom in the proposed EB network via a set of rules that guarantee eventual progress for higher-priority packets:

1. Each packet class has a dedicated VC at every router input port.
2. All arbiters enforce packet class priorities.
3. A router's scheduling of a low-priority packet never inhibits a subsequent high-priority packet from eventually reaching the first downstream EB.

In essence, a high priority packet must be able to advance from a VC, past the EB at a router's output port, and to the first downstream EB. From there, the packet can either proceed downstream if the channel is clear or dynamically allocate a VC at the router, switch to a new MECS channel, and advance by another hop. While the following discussion assumes two packet classes, the same reasoning applies to systems with more packet classes.

Together, the above rules allow the construction of an inductive proof showing that a high-priority packet will always be able to advance despite the presence of low-priority packets in the network. A Reply packet occupying a high-priority VC will eventually advance to at least the first downstream EB (rules 2,3). From the EB, it can acquire a VC at the associated router using JIT-VC (rules 1,2); buffer availability is guaranteed by virtue of another high-priority packet advancing by a hop (rules 2,3). Hop by hop, a high-priority packet will eventually reach its destination.

Additional care is required for handling two cases: (1) the first hop out of a node, and (2) transfers to the shared regions. First hop is challenging due to an EB at a router's output port, which offers no escape path (Figure 4). A reply can get stuck at this EB behind a request packet, violating

Table 2: Simulated network characteristics.

Network	1024 terminals with 256 concentrated nodes (64 shared resources), 128-bit links
Interconnect	Intermediate-layer wires: pitch = 100 nm, R = 8.6 kΩ/mm, C = 190 fF/mm
MECS (no PVC)	2 VCs/port, 35 flits/VC, 3 stage pipeline (VA-local, VA-global, XT)
MECS + PVC	25 VCs/port, 4 flits/VC, 3 stage pipeline (VA-local, VA-global, XT)
MECS + TAQ	Outside SR: conventional MECS w/o PVC. Within SR: MECS+PVC.
MECS + TAQ + EB	Outside SR: Per-class pure EB MECS networks: REQUEST (72 bits), REPLY (128 bits) 1 EB stage b/w adjacent routers, 2 stage pipeline (XA, XT), Within SR: MECS + PVC
K-MECS	Outside SR: single-network EB MECS with JIT-VC allocation, 1 EB stage b/w adjacent routers. Router: 2 VCs/port, 4 flits/VC, 2 stage pipeline (XA, XT), Within SR: MECS + PVC
Cmesh + PVC	6 VCs/port, 4 flits/VC, 2 stage pipeline (VA, XT)
common	XY dimension-order routing (DOR), VCT flow control, 1 injection VC, 2 ejection VCs
PVC QOS	400K cycles per frame interval
Workloads	Synthetic: *hotspot* and *uniform random* with 1- and 4-flit packets. PARSEC traces: see Table 3

Table 3: Simulated PARSEC traces.

Benchmark	Input Set	Simulated Cycles	Simulated Packets
blackscholes	small	255M	5.2M
blackscholes	medium	133M	7.5M
bodytrack	small	135M	4.7M
bodytrack	medium	137M	9.0M
canneal	medium	140M	8.6M
dedup	medium	146M	2.6M
ferret	medium	126M	2.2M
fluidanimate	small	127M	2.1M
fluidanimate	medium	144M	4.6M
swaptions	large	204M	8.8M
vips	medium	147M	0.9M
x264	small	151M	2.0M

Rule 3 above and potentially triggering deadlock. We resolve this condition by draining request packets into a low-priority VC at the first downstream node from a packet's source, allowing trailing packets to advance. The draining mechanism is triggered after a predetermined number of consecutive stall cycles at the first downstream EB and relies on JIT-VC allocation. To guarantee that a request packet can drain into an adjacent router, the switch allocator at the sending node checks for downstream buffer availability for each outbound request. If the allocator determines that buffer space may be unavailable by the time the request reaches the adjacent node, the packet is delayed.

Transfers to the shared region must also ensure destination buffer availability. The reason is that packets may escape blocked channels only through routers within their respective domain. Switching to a channel outside of a VM's domain violates the non-interference guarantee necessary for the topology-aware QOS architecture. Since transfers to the shared region (SR) may transit over multiple domains, buffer availability at an SR router must be guaranteed at the source to ensure that all SR-bound packets are eventually drained.

A single-network EB scheme described in this section enables a significant reduction in storage requirements for nodes outside of the shared regions. Assuming a maximum packet size of four flits and two priority classes, a pair of 4-deep VCs suffices at each router input port. Compared to a PVC-enabled MECS router with 25 VCs per port, both virtual channel and storage requirements are reduced by over 12x. Savings in storage requirements exceed 8x over a baseline MECS router with no QOS support.

4. EXPERIMENTAL METHODOLOGY

Area and energy: Our target configuration is a 1024-tile (256 node) CMP in 15 nm technology with on-chip voltage of 0.7 V. For both area and energy estimation, we use a combination of analytical models [12, 1], Orion [13], CACTI [23], previously published data [26], and synthesis results. We model a fixed chip area of 256 mm^2 and assume ideal dimension scaling of all devices and wires from 32 nm technology to arrive at our area estimates. We further assume fixed capacitance per unit length for both wires and devices to scale energy data from 0.9 V in 32 nm down to 0.7 V in 15 nm technology. We modify Orion to more accurately model crossbar fabrics, carefully accounting for the asymmetry in MECS, and apply segmentation [30] when profitable. In CACTI, we add support for modeling small SRAM FIFOs with data flow typical of a NOC router. We assume that VC FIFOs and PVC's flow state tables are SRAM-based. We estimate the energy consumption of an elastic buffer by synthesizing different primitive storage elements using a 45-nm technology library and extrapolate the results to our target technology. Transition probability for wires and logic is 0.5.

Channels: To reduce interconnect energy, we adopt a low-swing signaling scheme of Schinkel et al. [26]. The approach does not require a separate low-voltage power supply and supports low-overhead pipelined operation necessary for MECS. At 15 nm, low-swing wires improve energy-efficiency by 2.3x while reducing transceiver area by 1.6x versus full-swing interconnects. The area decreases due to elimination of repeaters required on full-swing links. Wire parameters are summarized in Table 2.

Network configurations: Network details are summarized in Table 2. Of the 256 network nodes, 64 correspond to shared resources. Configurations with topology-aware QOS support have four SR columns, with 16 shared resources per column. All networks utilize virtual cut-through flow control. We couple VC and crossbar allocation and perform switching at packet granularity to eliminate the need for a dedicated switch allocation stage. All configurations use look-ahead routing; PVC-enabled designs employ priority reuse [10]. These techniques remove routing and priority computation from the critical path. We model two packet sizes: 1-flit requests and 4-flit replies. Wire delay is one cycle between adjacent routers; channel width is 128 bits.

Baseline MECS: We model two baseline MECS networks – with and without PVC-based QOS support. Their respective VC configurations are described in Sec. 3.1.

MECS with TAQ: We evaluate a conventionally-buffered MECS network with the topology-aware QOS architecture. Routers inside the SRs are provisioned with PVC support, while the rest of the network features lighter-weight MECS routers with no QOS logic.

MECS with TAQ and dual-network EB: We augment the MECS+TAQ configuration with a pure elastic buffered flow control architecture [21]. The pure EB design eschews virtual channels, reducing router cost, but requires two networks – one per packet class. The Request network has a 72-bit datapath, while the Reply network has the full 128-bit width. Elastic buffering is deployed only outside the shared regions, with MECS+PVC routers used inside SRs. We do not evaluate an iDEAL organization [17], as it requires more buffer resources than our proposed approach and is therefore inferior in energy and area cost.

MECS with TAQ and single-network EB (K-MECS): Our proposed network architecture is called *Kilo-MECS (K-MECS)*. It combines TAQ with our single-network EB scheme, featuring elastic-buffered links, two VCs per router input port, and JIT-VC allocation.

Cmesh: We also evaluate a concentrated mesh (Cmesh) topology [1] due to its low area and wiring cost. Each PVC-enabled Cmesh router has six VCs per port and a single-stage VCT allocator. We do not consider a Cmesh+TAQ design, since a mesh topology is not compatible with topology-aware QOS organization.

Simulation-based studies: We use a custom NOC simulator to evaluate the performance and QOS impact of the various aspects of our proposal. We first examine the effect of individual techniques on performance and quality-of-service through focused studies on synthetic workloads. While these workloads are not directly correlated to expected traffic patterns of a CMP, they stress the network in different ways and provide insight into the effect of various mechanisms and topology options.

To evaluate parallel application network traffic, we used the M5 simulator [3] to collect memory access traces from a full system running PARSEC v2.1 benchmarks [2]. The simulated system is comprised of 64 two-wide superscalar out-of-order cores with private 32KB L1 instruction and data caches plus a shared 16MB L2 cache. Following the Netrace methodology [11], the memory traces are post-processed to encode the dependencies between transactions, which we then enforce during network simulation. Memory accesses are interleaved at 4KB page granularity among four on-chip memory controllers within network simulation. Table 3 summarizes the benchmarks used in our study. The benchmarks offer significant variety in granularity and type of parallelism. For each trace, we simulate no fewer than 100 million cycles of the PARSEC-defined region of interest (ROI).

5. EVALUATION RESULTS

We first evaluate the different network organizations on area and energy-efficiency. Next, we compare the performance of elastic buffered networks to conventionally buffered designs. We then discuss QOS implications of various topologies. Finally, we examine performance stability and QOS on a collection of trace-driven workloads.

5.1 Area

Our area model accounts for four primary components of area overhead: input buffers, crossbar switch fabric, flow state tables, and router-side elastic buffers. Results are shown in Figure 5(a). The MECS+EB and K-MECS* [†] bars corresponds to a router outside the shared region; all TAQ-enabled configurations use MECS+PVC routers inside the SR. We observe that elastic buffering is very effective in reducing router area in a MECS topology. Compared to a baseline MECS router with no QOS support, K-MECS* reduces router area by 61%. The advantage increases to 70% versus a PVC-enabled MECS router. A pure EB router (MECS+EB) has a 30% smaller footprint than K-MECS* for same datapath width; however, pure elastic buffering requires two networks, for a net loss in area efficiency.

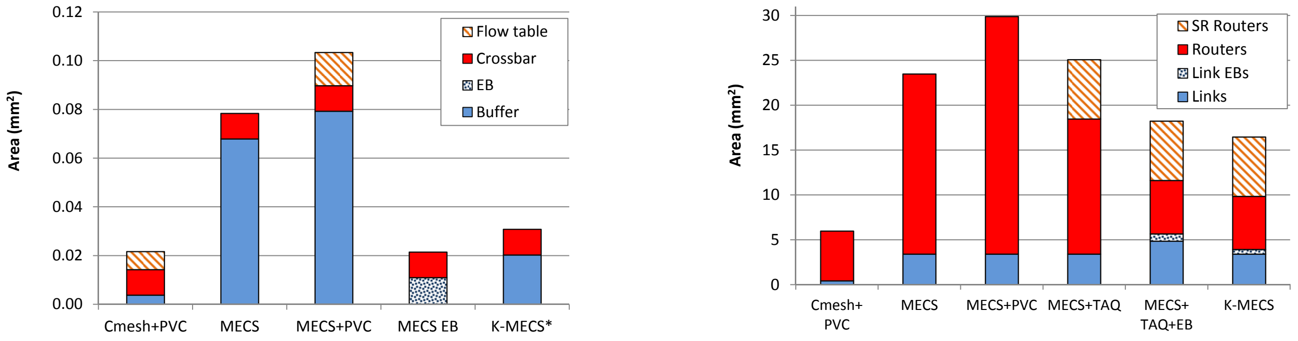
Figure 5(b) breaks down total network area into four resource types: links, link-integrated EBs, regular routers, and SR routers. The latter are applicable only to TAQ-enabled configurations. For links, we account for the area of drivers and receivers and anticipate that wires are routed over logic in a dedicated layer. TAQ proves to be an effective optimization for reducing network area. Compared to a conventionally-buffered MECS+PVC network, TAQ enables a 16% area reduction (MECS+TAQ bar). The pure elastic-buffered NOC further reduces the footprint by 27% (MECS+TAQ+EB) at the cost of a 56% increase in wire requirements. K-MECS offers an additional 10% area reduction without the extra wire expense by virtue of not requiring a second network. The conventionally-buffered SR routers in a K-MECS network make up a quarter of the network nodes yet account for over one-half of the overall router area. The smallest network area is found in the Cmesh topology due to its modest bisection bandwidth. The Cmesh NOC occupies 2.8 times less area than the K-MECS network but offers 8 times less network bandwidth.

5.2 Energy

Figure 6(a) shows the energy expended per packet for a router traversal in different topologies. As before, the MECS+EB and K-MECS* bars correspond to a router outside of the shared region, whereas the MECS+PVC datum is representative of an intra-SR router. Energy consumption in a K-MECS* router is reduced by 65% versus MECS with no QOS support and by 73% against a PVC-enabled MECS node. In addition to savings in buffer energy stemming from diminished storage requirements, K-MECS* also reduces switch energy relative to both MECS baselines. Reduction in switch energy is due to shorter input wires feeding the crossbar, which result from a more compact ingress layout. A pure EB router (MECS+EB) is 34% more energy efficient than K-MECS* by virtue of eliminating input SRAM FIFOs in favor of a simple double-latch elastic buffer and shorter wires feeding the crossbar.

In a Cmesh topology, a significant source of energy overhead is the flow state table required by PVC. In a mesh network, a large number of flows may enter the router from a single port, necessitating correspondingly large per-port state tables. In contrast, in a richly-connected MECS topology, flow state can be effectively distributed among the many input ports. Although the total required per-flow storage is

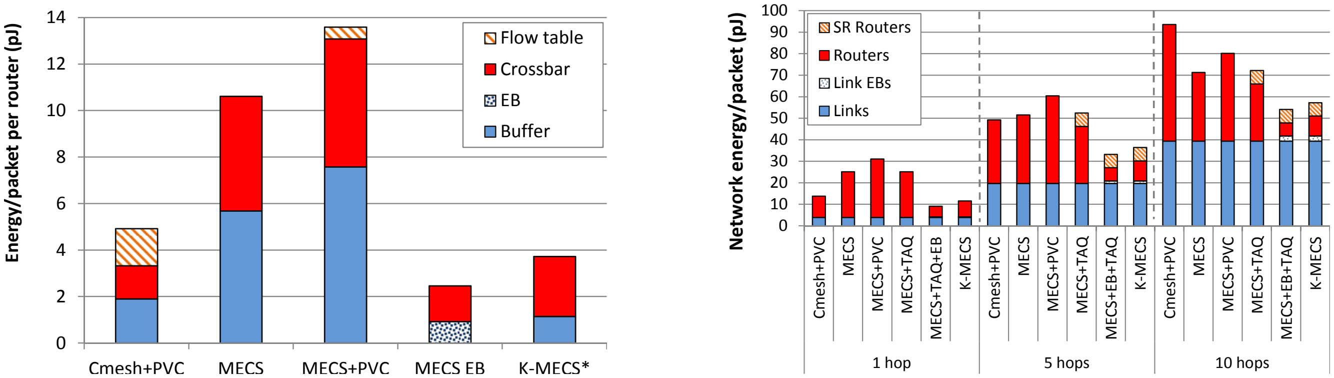
[†] We use K-MECS* to refer to the EB-enabled network outside of the shared regions. K-MECS refers to the entire heterogeneous NOC.



(a) Area of a single router

(b) Total network area

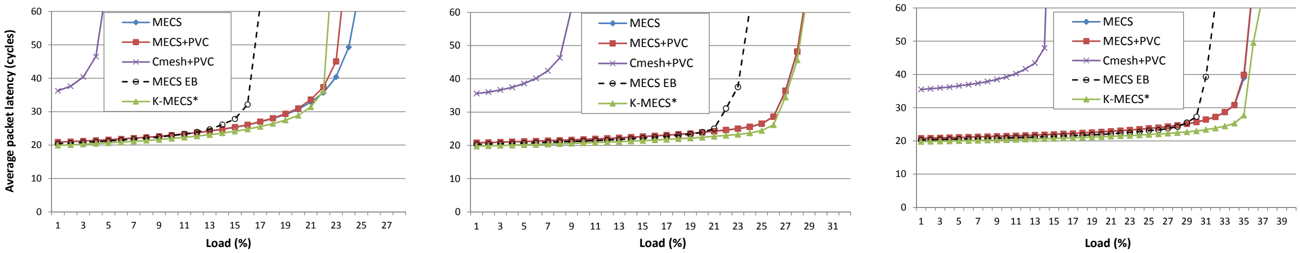
Figure 5: Router and network area efficiency.



(a) Router energy

(b) Network energy per packet

Figure 6: Router and network energy efficiency.



(a) 100% of terminals active

(b) 50% of terminals active

(c) 25% of terminals active

Figure 7: Performance comparison of different topologies for uniform random traffic.

comparable in Cmesh and MECS, the large physical tables in a Cmesh router incur a significant per-access energy penalty.

Figure 6(b) shows network-level energy efficiency for three different access patterns – nearest-neighbor (1-hop), semi-local (5 mesh hops), and random (10 mesh hops). The nearest-neighbor pattern incurs one link and two router traversals in all topologies. In contrast, 5-hop and 10-hop patterns are assumed to require three router accesses in the low-diameter MECS networks, while requiring 6 and 11 router crossings, respectively, in Cmesh. We assume that 25% of all accesses in the multi-hop patterns are to shared resources, necessitating transfers to and from the shared regions in TAQ-enabled networks.

In general, we observe that EB-enabled low-diameter networks have better energy efficiency than other topologies. A pure EB architecture is 22% more efficient than K-MECS on local traffic and 6-9% better on non-local routes thanks to a reduction in buffer and switch input power. K-MECS reduces NOC energy by 16-63% over remaining network architectures on local traffic and by 20-40% on non-local patterns. Links are responsible for a significant fraction of overall energy expense, diminishing the benefits of router energy optimizations. For instance, links account for 69% of the energy expanded on random traffic in K-MECS. PVC-enabled routers in the shared regions also diminish energy efficiency of K-MECS and other TAQ-enabled topologies.

Table 4: Fairness and throughput of different NOCs.

	min vs mean	max vs mean	std dev (% of mean)	throughput (% of max)
Cmesh	-100%	1009%	372%	89.7%
Cmesh+PVC	-9%	17%	5%	100%
MECS	-51%	715%	180%	100%
MECS+PVC	-1%	6%	1%	100%
K-MECS*	-52%	713%	181%	98.8%
K-MECS	-6%	5%	2%	100%

5.3 Performance

We evaluate the networks on a uniform random (UR) synthetic traffic pattern. This workload is highly sensitive to buffer capacity and is expected to challenge the storage-limited EB-enabled networks. We experiment with several different activity regimes for network nodes, noting that program phases and power constraints may limit the number of entities communicating at any one time. We report results for 100%, 50%, and 25% of terminals active. The active sources, if less than 100%, are chosen randomly at run time.

Figure 7 shows the results of the evaluation. Both EB configurations (MECS+EB and K-MECS*) model homogeneous NOCs without SRs to isolate the effect of elastic buffering on network performance. MECS+EB has dedicated request/reply networks. K-MECS* uses the JIT-VC allocation mechanism described in Section 3.3. In networks equipped with PVC, we disable the preemptive mechanism to avoid preemption-related throughput losses.

In general, low-diameter topologies with router-side buffering offer superior throughput over alternative organizations. With 100% of terminals communicating, K-MECS* shows a throughput loss of around 9% versus conventional MECS networks. Throughput is restored at 50% of the terminals utilized and slightly improves relative to the baseline when only 25% of the terminals are enabled. The improvement stems from the pipeline effect of EB channels which often allow packets to reach their destination despite downstream congestion. Without elastic buffering, a congested destination backpressures the source, causing head-of-line blocking at the injection port and preventing packets from advancing to less congested nodes.

The dual-network MECS+EB organization shows inferior performance versus other low-diameter designs despite a significant advantage in wire bandwidth. Compared to K-MECS*, throughput is reduced by 14-26% depending on the fraction of nodes communicating. Throughput suffers due to a lack of buffer capacity in pure EB routers, which backpressure into a MECS channel and block traffic to other nodes. Finally, the Cmesh network has the worst performance among the evaluated designs. Average latency at low loads is over 35 cycles per packet, a 1.8x slowdown relative to MECS. The high latency arises from the large average hop count of a mesh topology, while throughput is poor because of the low bisection bandwidth of the Cmesh network.

5.4 Quality-of-Service

To evaluate the fairness of various network configurations, we use a *hotspot* traffic pattern with a single hotspot node in the corner of the grid. We evaluate Cmesh, MECS, and K-MECS with and without PVC support. As before, K-MECS* represents a homogeneous organization with elastic buffering throughout the network and no QOS support. Table 4 summarizes the results of the experiment. The first two data columns show the minimum and maximum deviation from the mean throughput; a small deviation is desired, since it indicates minimal variance in throughput among the nodes. Similarly, the third data column shows the standard deviation from the mean; again, smaller is better. Finally, the last column plots overall network throughput with respect to the maximum achievable throughput in the measurement interval; in this case, higher is better since we seek to maximize throughput.

In general, all of the networks without QOS support are unable to provide any degree of fairness to the communicating nodes. In the CMesh network without PVC, many nodes are unable to deliver a single flit. In MECS and K-MECS*, the variance in throughput among the nodes is over 10x. PVC restores fairness. PVC-enabled MECS and K-MECS networks have a standard deviation from the mean of just 1-2%, with individual nodes deviating by no more than 6% from the mean throughput. Significantly, the proposed K-MECS organization with Topology-Aware QOS support is able to provide competitive fairness guarantees and good throughput while limiting the extent of hardware support to just a fraction of the network nodes.

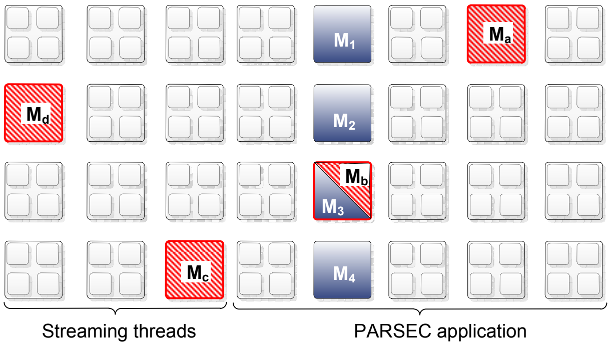


Figure 8: Trace-based evaluation setup. Memory controllers: shaded (K-MECS) and striped (MECS).

5.5 Trace-driven Evaluation

To assess the effectiveness of a topology-aware QOS architecture versus a conventional organization, we combine PARSEC trace-based workloads with synthetic traffic to model a denial-of-service attack in a multi-core CMP. We evaluate the architectures on their ability to provide application performance stability in the face of adverse network state. Figure 8 shows the experimental setup. We model a modestly-sized chip with 32 nodes, arranged in an 8x4 grid. On-chip memory controllers (MCs) occupy four nodes; remaining nodes are concentrated and integrate four core/cache terminals per node. Sixteen nodes are committed to a PARSEC application, while the remaining 12 continuously stream traffic to the memory controllers. Baseline MECS and CMesh network use a staggered memory controller placement, with MC locations striped and labeled M$_a$ through M$_d$ in the figure. The remaining NOCs employ a single shared region containing the four MC tiles, which are shaded and labeled M$_1$ through M$_4$ in the figure.

Figure 9 plots the slowdown of PARSEC packets in the presence of streaming traffic for the various network orga-

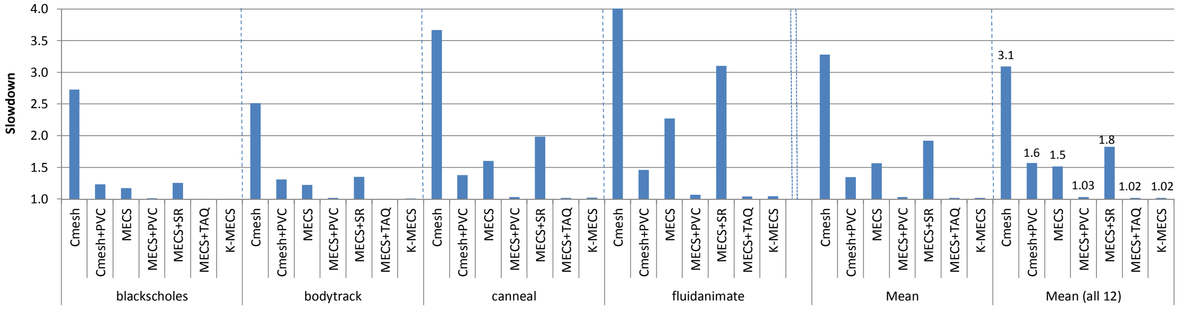


Figure 9: Average packet slowdown on PARSEC workloads with adversarial traffic.

nizations. We evaluate Cmesh and MECS topologies with staggered MCs (baseline) with and without PVC support. We also evaluate a MECS network with a shared region MC placement and PVC support inside the SR (MECS+TAQ). To isolate the benefits provided by the shared region organization, we introduce a MECS+SR variant that employs the SR but does not offer any QOS support. Finally, we evaluate the heterogeneous K-MECS organization that combines a conventionally-buffered PVC-enabled shared region with hybrid EB/VC buffering in the rest of the network.

Without QOS support, all networks suffer a performance degradation in the presence of streaming traffic. The degradation in MECS networks (MECS and MECS+SR) is less severe than in the CMesh NOC due to a degree of traffic isolation offered by a richly-connected MECS topology. Without QOS support, MECS+SR appears more susceptible to congestion than the baseline MECS organization. The latter is able to better tolerate network-level interference due to a more distributed MC placement.

PVC largely restores performance in all networks through improved fairness. Across the suite, all combinations of MECS and PVC result in a performance degradation of just 2-3%. MECS+TAQ, which relies on PVC only inside the shared region, shows the same performance resilience as the baseline MECS+PVC network. K-MECS is equally resilient, while using a fraction of the resources of other designs.

5.6 Summary

Table 5 summarizes the area, power requirements, and throughput of different topologies in a kilo-terminal network in 15 nm technology. Power numbers are derived for a 2 GHz clock frequency and random (10-hop) traffic described in Section 5.2. Throughput is for uniform random traffic with 50% of the nodes communicating. We observe that the proposed topology-aware QOS architecture is very effective at reducing network area and energy overhead without compromising performance. Compared to a baseline MECS network with PVC support, TAQ reduces network area by 16% and power consumption by 10% (MECS+TAQ). Furthermore, TAQ enables elastic buffered flow control outside of the shared regions that further reduces area by 27% and power draw by 25% but degrades throughput by over 17% (MECS+TAQ+EB). K-MECS combines TAQ with the single-network EB design also proposed in this work. The resulting organization restores throughput while improving area efficiency by yet another 10% with a small power penalty and no impact on QOS guarantees.

Table 5: Network area and power efficiency.

	Area (mm^2)	Power @ 1% (W)	Power @ 10% (W)	Max load (%)
Cmesh+PVC	6.0	3.8	38.3	9%
MECS	23.5	2.9	29.2	29%
MECS+PVC	29.9	3.3	32.9	29%
MECS+TAQ	25.1	3.0	29.6	29%
MECS+TAQ+EB	18.2	2.2	22.2	24%
K-MECS	16.5	2.3	23.5	29%

6. CONCLUSION

In this paper, we proposed and evaluated architectures for kiloscale networks-on-chip (NOC) that address area, energy, and quality-of-service (QOS) challenges for large-scale on-chip interconnects. We identify a low-diameter topology as a key Kilo-NOC technology for improving network performance and energy efficiency. While researchers have proposed low-diameter architectures for on-chip networks [16, 9], their scalability and QOS properties have not been studied. Our analysis reveals that large buffer requirements and QOS overheads stunt the ability of such topologies to support Kilo-NOC configurations in an area- and energy-efficient fashion.

We take a hybrid approach to network scalability. To reduce QOS overheads, we isolate shared resources in dedicated, QOS-equipped regions of the chip, enabling a reduction in router complexity in other parts of the die. The facilitating technology is a low-diameter topology, which affords single-hop interference-free access to the QOS-protected regions from any node. Our approach is simpler than prior network QOS schemes, which have required QOS support at every network node. In addition to reducing NOC area and energy consumption, the proposed topology-aware QOS architecture enables an *elastic buffering (EB)* optimization in parts of the network freed from QOS support. Elastic buffering further diminishes router buffer requirements by integrating storage into network links. We introduce a single-network EB architecture with lower cost compared to prior proposals. Our scheme combines elastic-buffered links and a small number of router-side buffers via a novel virtual channel allocation strategy.

Our final NOC architecture is heterogeneous, employing QOS-enabled routers with conventional buffering in parts of the network, and light-weight elastic buffered nodes elsewhere. In a kilo-terminal NOC, this design enables a 29% improvement in power and a 45% improvement in area over a state-of-the-art QOS-enabled homogeneous network at the

15 nm technology node. In a modest-sized high-end chip, the proposed architecture reduces the NOC area to under 7% of the die and dissipates 23W of power when the network carries a 10% load factor averaged across the entire NOC. While the power consumption of the heterogeneous topology bests other approaches, low-energy CMPs and SOCs will be forced to better exploit physical locality to keep communication costs down.

Acknowledgments

We wish to thank Naveen Muralimanohar, Emmett Witchel, and Andrew Targhetta for their contributions to this paper. This research is supported by NSF CISE Infrastructure grant EIA-0303609 and NSF grant CCF-0811056.

7. REFERENCES

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

[2] C. Bienia, S. Kumar, J. P. Singh, and K. Li. The PARSEC Benchmark Suite: Characterization and Architectural Implications. In *International Conference on Parallel Architectures and Compilation Techniques*, pages 72–81, October 2008.

[3] N. L. Binkert, R. G. Dreslinski, L. R. Hsu, K. T. Lim, A. G. Saidi, and S. K. Reinhardt. The M5 Simulator: Modeling Networked Systems. *IEEE Micro*, 26(4):52–60, July/August 2006.

[4] W. J. Dally. Virtual-channel Flow Control. In *International Symposium on Computer Architecture*, pages 60–68, June 1990.

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

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

[7] A. Demers, S. Keshav, and S. Shenker. Analysis and Simulation of a Fair Queueing Algorithm. In *Symposium Proceedings on Communications Architectures and Protocols (SIGCOMM)*, pages 1–12, September 1989.

[8] P. Gratz, C. Kim, K. Sankaralingam, H. Hanson, P. Shivakumar, S. W. Keckler, and D. Burger. On-Chip Interconnection Networks of the TRIPS Chip. *IEEE Micro*, 27(5):41–50, September/October 2007.

[9] B. Grot, J. Hestness, S. W. Keckler, and O. Mutlu. Express Cube Topologies for on-Chip Interconnects. In *International Symposium on High-Performance Computer Architecture*, pages 163–174, February 2009.

[10] B. Grot, S. W. Keckler, and O. Mutlu. Preemptive Virtual Clock: a Flexible, Efficient, and Cost-effective QOS Scheme for Networks-on-Chip. In *International Symposium on Microarchitecture*, pages 268–279, December 2009.

[11] J. Hestness, B. Grot, and S. W. Keckler. Netrace: Dependency-driven Trace-based Network-on-Chip Simulation. In *Workshop on Network on Chip Architectures*, pages 31–36, December 2010.

[12] International Technology Roadmap for Semiconductors. `http://www.itrs.net/links/2009ITRS/Home2009.htm`, 2009.

[13] A. Kahng, B. Li, L.-S. Peh, and K. Samadi. ORION 2.0: A Fast and Accurate NoC Power and Area Model for Early-stage Design Space Exploration. In *Design, Automation, and Test in Europe*, pages 423–428, April 2009.

[14] P. Kermani and L. Kleinrock. Virtual Cut-through: a New Computer Communication Switching Technique. *Computer Networks*, 3:267–286, September 1979.

[15] J. Kim. Low-cost Router Microarchitecture for On-chip Networks. In *International Symposium on Microarchitecture*, pages 255–266, December 2009.

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

[17] A. K. Kodi, A. Sarathy, and A. Louri. iDEAL: Inter-router Dual-Function Energy and Area-Efficient Links for Network-on-Chip (NoC) Architectures. In *International Symposium on Computer Architecture*, pages 241–250, June 2008.

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

[19] J. W. Lee, M. C. Ng, and K. Asanović. Globally-Synchronized Frames for Guaranteed Quality-of-Service in On-Chip Networks. In *International Symposium on Computer Architecture*, pages 89–100, June 2008.

[20] M. R. Marty and M. D. Hill. Virtual Hierarchies to Support Server Consolidation. In *International Symposium on Computer Architecture*, pages 46–56, June 2007.

[21] G. Michelogiannakis, J. Balfour, and W. Dally. Elastic-buffer Flow Control for On-chip Networks. In *International Symposium on High-Performance Computer Architecture*, pages 151 –162, February 2009.

[22] T. Moscibroda and O. Mutlu. A Case for Bufferless Routing in On-Chip Networks. In *International Symposium on Computer Architecture*, pages 196–207, 2009.

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

[24] NVIDIA. NVIDIA's Next Generation CUDA Compute Architecture: Fermi. `http://www.nvidia.com/content/PDF/fermi_white_papers/NVIDIA_Fermi_Compute_Architecture_Whitepaper.pdf`, 2009.

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

[26] D. Schinkel, E. Mensink, E. Klumperink, E. van Tuijl, and B. Nauta. Low-Power, High-Speed Transceivers for Network-on-Chip Communication. *IEEE Transactions on VLSI Systems*, 17(1):12 –21, January 2009.

[27] J. Shin, K. Tam, D. Huang, B. Petrick, H. Pham, C. Hwang, H. Li, A. Smith, T. Johnson, F. Schumacher, D. Greenhill, A. Leon, and A. Strong. A 40nm 16-core 128-thread CMT SPARC SoC Processor. In *International Solid-State Circuits Conference*, pages 98–99, February 2010.

[28] Tilera TILE-Gx100. `http://www.tilera.com/products/TILE-Gx.php`.

[29] 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.

[30] H. Wang, L.-S. Peh, and S. Malik. Power-driven Design of Router Microarchitectures in On-chip Networks. In *International Symposium on Microarchitecture*, pages 105–116, December 2003.

[31] D. Wentzlaff, P. Griffin, H. Hoffmann, L. Bao, B. Edwards, C. Ramey, M. Mattina, C.-C. Miao, J. F. B. III, and A. Agarwal. On-Chip Interconnection Architecture of the Tile Processor. *IEEE Micro*, 27(5):15–31, September/October 2007.