The Dissertation Committee for Boris Grot
certifies that this is the approved version of the following dissertation:

Network-on-Chip Architectures for Scalability and Service Guarantees

Committee:

Stephen W. Keckler, Supervisor

Douglas C. Burger

Onur Mutlu

Emmett Witchel

Yin Zhang

Network-on-Chip Architectures for Scalability and Service Guarantees

by

Boris Grot, B.S.; M.S.E.E.

Dissertation

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of Philosophy

The University of Texas at Austin

August 2011

To my wife, Gaya,

and my loving parents

Acknowledgments

As I finish this journey, my heart fills with gratitude for the many people who made it possible. First and foremost, I am forever indebted to my parents, without whom I would not be here. My mother, Irena, made numerous sacrifices in my name, of which our immigration from the Soviet Union is just one example. My step-father, Yume, has been my role model and a constant source of inspiration, thanks to his thirst for knowledge, love of life, and a philosophical approach to everything.

I extend the deepest appreciation to my advisor, Steve Keckler. As a scientist, Steve constantly challenged me to look deeper and question the assumptions. As a mentor, he wholeheartedly supported my aspirations and provided me with invaluable opportunities for growth. As a manager, he taught me the power of respect. I am especially grateful to Steve for letting me find my own path as he provided the nurturing blend of support, encouragement, patience, and funding.

I was very fortunate to work with, and learn from, a number of incredible people at UT. I particularly want to thank Doug Burger for the profound influence he has had on me through his combination of vision, scientific courage, and wit. Professors Kathryn McKinley and Emmett Witchel have been valuable sources of technical knowledge and tactical advice. Many fellow graduate

students provided cognitive fodder, critique, and emotional support. In particular, Karu Sankaralingam and Simha Sethumadhavan served as my academic role models, offering much-appreciated advice and encouragement. Paul Gratz and Joel Hestness were my fantastic and tireless collaborators. CARTelians Renee St. Amant, Katie Coons, Jeff Diamond, Hadi Esmaeilzadeh, Mark Gebhart, Sibi Govindan, Heather Hanson, Dong Li, Haiming Liu, Bert Maher, and Behnam Robatmili attended my practice talks, proof-read papers, participated in discussions, listened, and argued like good friends should. A warm thanks to Mike Bond, Milind Kulkarni, Phoebe Nethercote, and Ben Wiedermann for their friendship and lunch-time diversions.

My graduate career began at UCLA, where I took first steps in computer architecture research under the guidance of Bill Mangione-Smith and Glenn Reinman. I am very grateful to Bill for having faith in me and getting me started on a path to becoming an independent researcher. I thank Glenn for his unconditional support and encouragement, as well as for facilitating my move to UT. Those years would not have been as great as they were without my friends: Stephen Brown, Mehul Shah, Dan Shiber, Steve Wedig, and Yulia Zabiyaka.

Many others poured their time, energy, and resources into me. Onur Mutlu provided me with the freedom to pursue my own research agenda during a summer internship at Microsoft Research and had since become a very supportive collaborator. Gem Naivar, Gloria Ramirez, and Lydia Griffith dispensed the magic combination of administrative support and personal attention to virtually eliminate the bureaucratic pain of graduate school. DARPA and NSF generously sponsored my doctoral research through grants.

Graduate school would be a bleak affair without the warm support of my friends. I am deeply indebted to my tango family, in Austin and elsewhere, for the warm hugs in the toughest of days and the darkest of nights. I am grateful to my Russian-speaking circle, friends and relatives alike, in Austin, Boston, and Los Angeles – you are my link to the homeland. A special thanks to Anatoliy Eybelman and Elena Lande for their lasting friendship.

Above all, I wish to thank my wife, Gaya. Thank you for your faith in me, your care for me, and, most importantly, your undying love and understanding.

Boris Grot

The University of Texas at Austin
August 2011

Network-on-Chip Architectures for Scalability and Service Guarantees

Publication No. ____________________

Boris Grot, Ph.D.
The University of Texas at Austin, 2011

Supervisor: Stephen W. Keckler

Rapidly increasing transistor densities have led to the emergence of richly-integrated substrates in the form of chip multiprocessors and systems-on-a-chip. These devices integrate a variety of discrete resources, such as processing cores and cache memories, on a single die with the degree of integration growing in accordance with Moore's law. In this dissertation, we address challenges of scalability and quality-of-service (QOS) in network architectures of highly-integrated chips. The proposed techniques address the principal sources of inefficiency in networks-on-chip (NOCs) in the form of performance, area, and energy overheads. We also present a comprehensive network architecture ca-

pable of interconnecting over a thousand discrete resources with high efficiency and strong guarantees.

We first show that mesh networks, commonly employed in existing chips, fall significantly short of achieving their performance potential due to transient congestion effects that diminish network performance. Adaptive routing has the potential to improve performance through better load distribution. However, we find that existing approaches are myopic in that they only consider local congestion indicators and fail to take global network state into account. Our approach, called Regional Congestion Awareness (RCA), improves network visibility in adaptive routers via a light-weight mechanism for propagating and integrating congestion information. By leveraging both local and non-local congestion indicators, RCA improves network load balance and boosts throughput. Under a set of parallel workloads running on a 49-node substrate, RCA reduces on-chip network latency by 16%, on average, compared to a locally-adaptive router.

Next, we target NOC latency and energy efficiency through a novel point-to-multipoint topology. Ring and mesh networks, favored in existing on-chip interconnects, often require packets to go through a number of intermediate routers between source and destination nodes, resulting in significant latency and energy overheads. Topologies that improve connectivity, such as fat tree and flattened butterfly, eliminate much of the router overhead, but require non-minimal channel lengths or large channel count, reducing energy-efficiency and/or performance as a result. We propose a new topology, called Multidrop Express Channels (MECS), that augments minimally-routed express channels with multi-drop capability. The resulting richly-connected NOC enjoys a low hop count with favorable delay and energy characteristics, while

improving wire utilization over prior proposals.

Applications such as virtualized servers-on-a-chip and real-time systems require chip-level quality-of-service (QOS) support to provide fairness, service differentiation, and guarantees. Existing network QOS approaches suffer from considerable performance and area overheads that limit their usefulness in a resource-limited on-die network. In this dissertation, we propose a new QOS scheme called Preemptive Virtual Clock (PVC). PVC uses a preemptive approach to provide hard guarantees and strong performance isolation while dramatically reducing queuing requirements that burden prior proposals.

Finally, we introduce a comprehensive network architecture that overcomes the bottlenecks of earlier designs with respect to area, energy, and QOS in future highly-integrated chips. The proposed NOC uses a topology-centric QOS approach that restricts the extent of hardware QOS support to a fraction of the network without compromising guarantees. In doing so, network area and energy efficiency are significantly improved. Further improvements are derived through a novel flow-control mechanism, along with switch- and link-level optimizations. In concert, these techniques yield a network capable of interconnecting over a thousand terminals on a die while consuming 47% less area and 26% less power than a state-of-the-art QOS-enabled NOC.

The mechanisms proposed in this dissertation are synergistic and enable efficient, high-performance interconnects for future chips integrating hundreds or thousands of on-die resources. They address deficiencies in routing, topologies, and flow control of existing architectures with respect to area, energy, and performance scalability. They also serve as a building block for cost-effective advanced services, such as QOS guarantees at the die level.

Contents

Acknowledgments **v**

Abstract **viii**

Contents **xi**

List of Tables **xvi**

List of Figures **xvii**

Chapter 1 Introduction **1**

1.1 Networks-on-Chip 2

1.1.1 Motivation 2

1.1.2 Technology-Driven Network Design 3

1.1.3 Limitations of Existing NOC Architectures 4

1.2 Thesis Statement 5

1.3 Dissertation Contributions 6

1.4 Dissertation Organization 8

Chapter 2 Background **9**

2.1 Fundamentals of Interconnection Networks 9

2.2 Router Microarchitecture 12

2.3 Evaluation Metrics 14

2.3.1 Performance 14

2.3.2 Area . . . 14
2.3.3 Energy . . . 16
2.3.4 Quality-of-Service . . . 17
2.4 Summary . . . 18

Chapter 3 Regional Congestion Awareness 19
3.1 Background and Prior Work . . . 21
3.1.1 Routing Policies . . . 21
3.1.2 Congestion Management . . . 23
3.2 Network-on-Chip Adaptive Routers . . . 24
3.2.1 Adaptive Router Microarchitecture . . . 24
3.2.2 Local Contention Metrics . . . 26
3.3 Regional Congestion Awareness . . . 27
3.3.1 RCA Variants . . . 28
3.3.2 RCA Microarchitecture . . . 29
3.3.3 Status Network Design . . . 33
3.4 Evaluation . . . 34
3.4.1 Methodology . . . 34
3.4.2 Workloads . . . 35
3.4.3 Evaluation of RCA Metrics . . . 36
3.4.4 Sensitivity to Network Design Point . . . 40
3.4.5 Discussion . . . 43
3.5 Summary . . . 44

Chapter 4 Express Cube Topologies for On-Chip Interconnection Networks 46
4.1 Background . . . 48
4.2 Multidrop Express Channels . . . 50
4.2.1 Overview . . . 50
4.2.2 Microarchitecture . . . 51
4.2.3 Analysis . . . 54
4.2.4 Multicast and Broadcast . . . 57

4.2.5 Comparison to Prior Work 57
4.3 Generalized Express Cubes 58
4.4 Experimental Methodology 61
4.5 Evaluation 66
4.5.1 Synthetic Workload: 64 terminals 66
4.5.2 Synthetic Workload: 256 terminals 69
4.5.3 Area 71
4.5.4 Energy 74
4.5.5 Area-normalized Topology Comparison 77
4.5.6 Application-based Workloads 78
4.5.7 Summary 79
4.6 Discussion 80
4.6.1 VC Sensitivity 80
4.6.2 Switch Connectivity 84
4.6.3 Adaptive Routing 85
4.7 Summary 87

Chapter 5 Preemptive Virtual Clock: A Flexible, Efficient, and Cost-effective QOS Scheme for Networks-on-Chip 88

5.1 Motivation 90
5.1.1 NOC QOS Requirements 90
5.1.2 QOS Service Disciplines 91
5.2 Preemptive Virtual Clock 96
5.2.1 Overview 97
5.2.2 QOS Particulars 100
5.2.3 Microarchitecture 103
5.2.4 Comparison to Prior Approaches 106
5.3 Methodology 108
5.4 Evaluation 111
5.4.1 Quality-of-Service 111
5.4.2 Throughput and Performance Scalability 113

5.4.3 Performance Isolation 115
5.4.4 Differentiated Services 118
5.4.5 Energy 119
5.4.6 Storage Requirements 120
5.5 Discussion 121
5.5.1 NOC Flow Control Mechanisms 121
5.5.2 Effect of Flow Control on Fairness 123
5.5.3 Effect of Flow Control and Topology on Preemption Incidence 124
5.6 Conclusion 128

Chapter 6 Kilo-NOC: A Heterogeneous Network-on-Chip Architecture for Scalability and Service Guarantees 130
6.1 NOC Scalability Bottlenecks 132
6.2 Kilo-NOC Architecture 136
6.2.1 Baseline Design 136
6.2.2 Topology-aware QOS Architecture 138
6.2.3 Low-Cost Elastic Buffering 141
6.3 Experimental Methodology 146
6.4 Evaluation Results 150
6.4.1 Area 151
6.4.2 Energy 152
6.4.3 Performance 155
6.4.4 Quality-of-Service 157
6.4.5 Trace-driven Evaluation 158
6.4.6 Summary 161
6.5 Conclusion 161

Chapter 7 Conclusions 163
7.1 Dissertation Summary 164
7.1.1 Congestion-Aware Routing 164
7.1.2 Low-Diameter NOC Topologies 166

7.1.3 Preemptive QOS Architecture 167

7.1.4 Kilo-node NOC Architectures 169

7.2 Concluding Thoughts 170

7.2.1 Limitations of Proposed Techniques 170

7.2.2 Future Challenges and Opportunities 173

Bibliography **176**

Vita **188**

List of Tables

3.1 Baseline network configuration and variations. 35

4.1 Comparison of Concentrated Mesh (Cmesh), Flattened Butterfly, and MECS topologies. 54
4.2 Simulated network configurations. 63
4.3 Full-system configuration. 64

5.1 Feature comparison of QOS schemes. 106
5.2 Simulation methodology details. 109
5.3 *Relative throughput* of different QOS schemes. 112
5.4 Packet delay variation (jitter) of different QOS schemes. . . . 112
5.5 Differential bandwidth allocation in PVC. 118
5.6 Per-node storage requirements of different QOS schemes. . . . 120
5.7 *Relative throughput* under flit- and packet-level flow control in a 64-terminal PVC network. 123

6.1 Scalability of NOC topologies. 133
6.2 Simulated network characteristics. 148
6.3 Simulated PARSEC traces. 150
6.4 Fairness and throughput of different NOCs. 158
6.5 Network area and power efficiency. 161

List of Figures

1.1 Two-dimensional *k-ary n-cube* topologies 4

2.1 Canonical NOC router microarchitecture. 13

3.1 Taxonomy of routing policies with respect to congestion avoidance. 22
3.2 Microarchitecture of a two-stage locally-adaptive router. . . . 25
3.3 Regional Congestion Awareness variants. 28
3.4 Microarchitecture of an RCA router. 30
3.5 RCA microarchitectural details. 31
3.6 Performance comparison of RCA, locally adaptive, and oblivious routing. 37
3.7 Average latency for 100 permutations of random pairs traffic. . 38
3.8 Average latency across SPLASH-2 benchmarks normalized to latency of DOR. 39
3.9 Performance in 4x4 and 16x16 meshes on *bit-complement* traffic. 40
3.10 Performance with short and long packets on *bit-complement* traffic. 41
3.11 Performance with four virtual channels on *bit-complement* traffic. 42

4.1 Concentrated Mesh, Flattened Butterfly and MECS topologies in a 64-terminal network. 49
4.2 MECS router microarchitecture. 52
4.3 Detailed MECS router microarchitecture. 53

4.4 MECS variants for cost-performance trade-off. 60
4.5 Performance of different topologies in a 64-terminal NOC. . . 67
4.6 Performance of different topologies in a 256-terminal NOC. . . 70
4.7 Area requirements of different NOC organizations under fixed bisection wire budget. 72
4.8 Energy per transaction in different NOC organizations. 75
4.9 Topology comparison under a fixed 10 mm^2 area budget. . . . 77
4.10 Performance and energy efficiency of different topologies on the PARSEC suite. 79
4.11 VC sensitivity of low-diameter topologies in a 64-terminal NOC. 81
4.12 Sensitivity of low-diameter topologies to injection VC count. . 83
4.13 Sensitivity of MECS to crossbar connectivity. 85
4.14 Performance of routing policies and topologies on *transpose* traffic. 86

5.1 Scenario demonstrating poor bandwidth utilization with GSF. 94
5.2 Performance of GSF with various frame and window sizes. . . 95
5.3 Comparison of GSF and PVC framing strategies. 98
5.4 PVC router microarchitecture. 104
5.5 Performance of WFQ, GSF and PVC on uniform random traffic. 114
5.6 Experimental setup for PARSEC workloads. 116
5.7 Experimental results on PARSEC workloads. 117
5.8 PVC energy overhead over a generic NOC with no QOS support. 119
5.9 Preemption incidence in mesh, Cmesh, and MECS topologies with flit- and packet-level flow control in a 64-terminal network. 125
5.10 Preemption incidence in mesh, Cmesh, and MECS topologies with packet-level flow control in a 256-terminal network. . . . 127

6.1 64-tile CMP with 4-way concentration and MECS topology. . 137
6.2 Elastic buffer deadlock avoidance. 143
6.3 MECS with deadlock-free elastic buffer. 144
6.4 Router and network area efficiency. 152
6.5 Router and network energy efficiency. 154

6.6 Performance comparison of different topologies on random traffic. 156
6.7 Experimental setup for PARSEC trace-based evaluation. . . . 159
6.8 Average packet slowdown on PARSEC workloads with adversarial traffic. 160

Chapter 1

Introduction

Over the past 40 years, the semiconductor industry has delivered continuous improvements in transistor density, approximately doubling the number of devices integrated on a die in every technology generation. As a result, starting with the *Intel 4004* single-chip CPU introduced in 1971 with 2300 transistors, on-die device counts have grown by six orders of magnitude to over 2.3 billion in the 2010 Intel *Xeon Nehalem-EX*.

For much of this time, increasing transistor budgets were largely aimed at boosting single-threaded performance, which improved by as much as 50-60% per year through the 1990's [3]. Performance gains were derived through a combination of increased microarchitectural complexity and higher clock rates. In the early 2000's, however, a combination of factors that include design complexity considerations, growing wire delays, and power limitations of densely-integrated chips started limiting gains in core-level performance. Since then, core complexity has remained relatively constant, clock rates have stagnated, and growing transistor budgets have been devoted to additional cores, memories, and other on-chip resources.

Today, chip-level multiprocessors (CMPs) and systems-on-a-chip (SOCs) are commonly found in a broad range of applications. These devices commonly integrate a combination of general-purpose cores, specialized hardware accelerator engines, caches, software-controlled memories, DRAM controllers, and I/O interfaces. Contemporary examples of such chips include a 16-core server processor from Sun/Oracle [66], an 80-core research prototype from Intel [77], and a 100-core processor from Tilera [74]. Trends point in the direction of further integration, and chips containing thousands of cores and other resources are anticipated in the future.

1.1 Networks-on-Chip

1.1.1 Motivation

Chip multiprocessors place considerable demands on their die-level interconnects. These demands are driven by technology constraints, application characteristics, and market forces. Specifically, on-chip constraints mandate low area and energy footprints; applications require high network performance; and businesses demand advanced features such as chip-level service guarantees.

To interconnect the various on-chip resources, early CMPs employed bus- and crossbar-based interconnect fabrics [46]. While acceptable for chips with a small number of discrete components, these interconnects proved challenging to scale to configurations with tens of on-die resources. In response, researchers proposed structured, packet-based, multi-hop networks-on-chip (NOCs) [14, 7] that enjoy better scalability properties than buses and crossbars. Both research and commercial multicore processors employing ring- and mesh-based NOCs have since appeared [57, 77, 74].

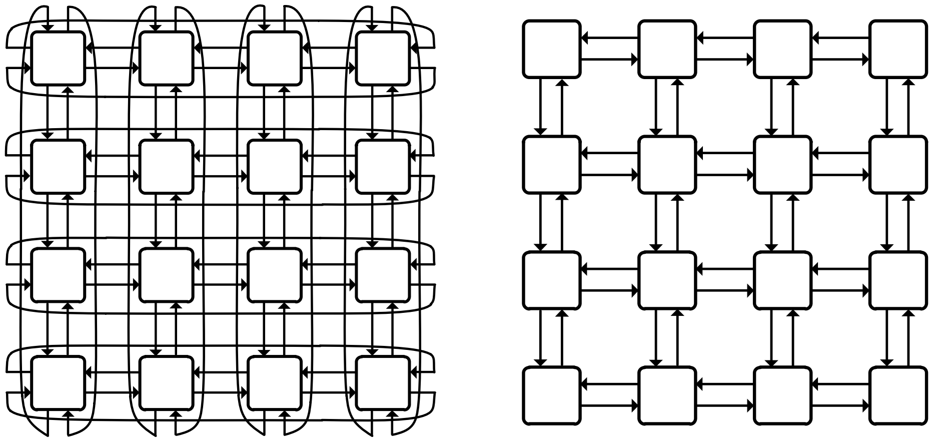
At a high level, on-chip networks are similar to their off-chip counterparts. Both rely on a distributed interconnect fabric composed of data-carrying links, also referred to as channels, and routers that provide buffering and switching functionality. However, significant differences in technology constraints necessitate different organizations for on- and off-chip networks. In the off-chip domain, the primary determinant of network performance and cost is the available pin budget [15]. Networks-on-chip are free of pin constraints, and, thanks to rich wire resources available on a die, enjoy a tremendous bandwidth advantage over chip-to-chip interconnects. On the other hand, the design space of NOCs is more constrained relative to that of off-chip networks due to the limited area and power envelope of highly-integrated CMPs.

1.1.2 Technology-Driven Network Design

To understand the implications imposed by the technology constraints on network design, consider the following case study. Since the late 80's, a popular topology for high-performance computer interconnects has been a two- or three-dimensional *torus*, successfully employed in supercomputers from the likes of Cray [64] and IBM [25]. Figure 1.1(a) shows a toy 4x4 network connected via a 2-D torus topology. A salient feature of this network is its low dimensionality, meaning that each router has few connections to other nodes (four connections in the case of a 2-D torus and six in a 3-D organization). From the stand-point of a pin-limited router chip, low connectivity is attractive as it affords wide interfaces that reduce the packet serialization latency.

One concern with a low-dimensional topology is that it may require a number of router crossings for packets navigating a large network. Each router traversal is a source of additional delay, which can negate the latency-reducing benefits of wide channels. A simple analysis, however, shows that channel delay in off-chip networks greatly dominates the latency required to traverse a router. Signal velocity in a copper or fiber optic cable is within two-thirds of the speed of light in vacuum, or around 200,000 m/s. Assuming a channel 10 m in length, the wire flight time is 50 ns. In comparison, an Alpha 21364 router designed for a 2-D torus network and operating at 1.2 Ghz had a pin-to-pin delay of just 10.8 ns [49]. Such a router would contribute under 22% to the delay of a 10 m long wire, which represents a modest performance overhead.

In contrast, die dimensions of even the high-end chips rarely exceed 25 mm per edge. Typical communication distances are on the order of a



(a) Torus

(b) Mesh

Figure 1.1: Two-dimensional *k-ary n-cube* topologies

few millimeters, reducing the wire flight time to a few nanoseconds even over highly-resistive on-chip wires. In substrates with such short communication distances, routers constitute a significant fraction of network delay and energy overhead. This attribute reduces the advantages of low-dimensional topologies in an on-die setting despite their appeal in off-chip networks.

1.1.3 Limitations of Existing NOC Architectures

On-die constraints demand NOC architectures that offer high energy efficiency, small area footprint, modest wiring complexity, low communication latency, and good throughput under load. This combination of demands is difficult to satisfy through existing interconnect architectures that may meet some of these objectives while failing on the rest.

For instance, a number of existing chips developed in both industry and academia implement a two-dimensional *mesh* topology [78, 60, 74, 77], shown in Figure 1.1(b). The mesh is closely related to the torus and comes from the same *k-ary n-cube* family of topologies. It enjoys the same nearest-neighbor connectivity as the torus but without the wrap-around links that double the wire costs. The mesh organization enjoys low router complexity, short channel lengths, and planar layout that is die-friendly. However, due to the short communication distances on a chip, even light-weight routers constitute a significant latency overhead. In addition, each router crossing carries an energy cost due to the need to access packet buffers and traverse a switch fabric. These features make the mesh, along with other topologies with low connectivity, a poor fit in richly-integrated chips that are often sensitivity to both energy and performance.

One way to improve network performance and energy-efficiency is through the use of NOC architectures with richer connectivity among the nodes. By bypassing some number of intermediate routers between a packet's source and destination, richly-connected topologies reduce the routers' contribution to network latency and energy overhead. However, these benefits come at a price. Specifically, richly-connected networks require significant wiring resources and necessitate complex, high-radix routers that may increase network area footprint and add delay in near-neighbor communication patterns.

In addition to the considerations above, features such as adaptive routing and quality-of-service support tend to grow router complexity and may increase node delay, area, and energy costs regardless of the topology. These observations motivate scalable NOC architectures that account for chip-level constraints while leveraging the strengths of an on-die implementation.

1.2 Thesis Statement

This dissertation addresses the design of on-chip networks that satisfy performance, efficiency, and quality-of-service demands of highly-integrated CMPs and SOCs. Specifically, we propose (1) a congestion-aware routing mechanism that boosts network performance with no sacrifice in efficiency; (2) a richly-connected topology that reduces network delay and energy overheads at low router and wiring cost; (3) a quality-of-service architecture with strong bandwidth guarantees at modest efficiency overheads; and (4) a highly scalable NOC architecture that combines and extends the above techniques with QOS and flow control improvements to enable kilo-node on-die networks with good performance, high efficiency, and strong guarantees.

1.3 Dissertation Contributions

This dissertation makes the following contributions:

Congestion-Aware Routing: We introduce a family of policies and light-weight microarchitectural support for network-wide congestion notification. Our approach, called Regional Congestion Awareness (RCA), enhances conventional adaptive routers that rely on local-only congestion estimates with broader knowledge of network state. RCA relies on a specialized reduction network that aggregates local and non-local congestion indicators at each network routers and propagates the results to adjacent nodes. The reduction approach scales naturally to large NOC sizes and carries minimal logical and wiring expense. By disseminating congestion indicators throughout the network, RCA empowers adaptive routers to make better-informed decisions and improves the distribution of traffic over NOC links. Our evaluation confirms this claim and shows that an RCA-enabled NOC can boost throughput and reduce packet latency over locally-adaptive routers through improved network load balance.

Multidrop Express Channels Topology: To reduce network energy consumption and communication latency, this dissertation proposes a low-network-diameter Multidrop Express Channels (MECS) topology. A distinguishing feature of MECS is its use of point-to-multipoint channels that enable rich connectivity in a bandwidth-efficient manner. Compared to mesh-based organizations, MECS reduces the number of router traversals on traffic with limited or no locality. As routers are responsible for a considerable fraction of network energy and delay overhead, MECS improves both performance and energy-efficiency. Contrasted with existing low-diameter networks which require dedicated point-to-point channels between interconnected routers, MECS reduces the number of required channels and diminishes router complexity. These features improve network performance and scalability.

Preemptive Virtual Clock QOS Architecture: Existing CMPs and SOCs lack a way to enforce priorities, provide isolation, and satisfy QOS demands in the face of fine-grained resource sharing by disparate threads sharing a substrate. In response, we develop a light-weight QOS scheme called Preemptive Virtual Clock (PVC). PVC provides performance isolation, strong bandwidth guarantees, and good performance without the overheads of prior approaches. PVC features low scheduling complexity and minimizes buffer requirements through the use of preemption to overcome in-network priority inversion. By controlling preemption aggressiveness, PVC enables a trade-off between the strength of the guarantees and network performance. PVC also simplifies network management through a flexible allocation mechanism that enables per-application or per-VM bandwidth provisioning independent of thread count and supports transparent bandwidth recycling among a group of threads.

Kilo-NOC: A Scalable, Heterogeneous NOC: Finally, to support the integration requirements of future chips, this thesis proposes and evaluates technologies to enable networks-on-chip to support over a thousand connected components (Kilo-NOC) with high area and energy efficiency, good performance, and strong QOS guarantees. We identify buffer requirements of low-diameter topologies and QOS overheads as chief scalability obstacles in kilo-node NOCs under technology scaling. In response, we propose a 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 elegantly combines virtual channel and elastic buffer flow control. 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.

1.4 Dissertation Organization

The remainder of this dissertation is organized as follows. Chapter 2 presents the fundamental concepts in interconnection networks, introduces the principal metrics for evaluating these networks, and describes the baseline NOC router microarchitecture. Chapter 3 addresses congestion-aware routing with Regional Congestion Awareness. Chapter 4 describes Multidrop Express Channels, a low-diameter bandwidth-efficient NOC topology. Chapter 5 motivates the need for on-chip quality-of-service support and introduces Preemptive Virtual Clock, a low-cost NOC QOS scheme. Chapter 6 proposes a scalable NOC architecture that combines the proposed technologies with several additional optimizations to efficiency interconnect over a thousand nodes on a die. Chapter 7 summarizes the key contributions of this dissertation and concludes with thoughts on future research directions.

Chapter 2

Background

In this chapter, we review the fundamental concepts of interconnection networks, describe a canonical NOC router architecture, and discuss the metrics for analyzing on-chip interconnects.

2.1 Fundamentals of Interconnection Networks

Historically, interconnection networks have been classified along three dimensions: *topology*, *routing*, and *flow control* [15]. We define *quality-of-service* as the fourth dimension for characterizing on-chip interconnect fabrics due to the necessity of supporting resource sharing among concurrently executing applications or virtual machines. Here, we briefly review the chief attributes of these dimensions and summarize their implications for on-chip networks. A careful treatment of each topic, along with related prior work, can be found in Chapters 3 (routing), 4 (topology), and 5 (flow control and quality-of-service).

Topology

Network topology reflects the arrangement of nodes in an interconnection network and the connectivity among them. On a two-dimensional die, nodes are naturally arranged in a plane, often in a grid-like fashion. *Network diameter* is defined as the maximum number of hops, or router crossings, in a given network. Topologies with limited connectivity, such as rings and meshes, may require a packet to cross a number of routers on its way to the destination; such topologies are said to have a large network diameter. In contrast, *low-diameter* networks have rich connectivity that reduce the number of hops for a typical packet transfer. While rich connectivity is clearly an advantage, low-diameter networks may require higher router complexity, wire bandwidth, and design effort as compared to organizations with limited connectivity. The challenge for NOC architects is thus in designing richly-connected on-chip topologies that are also complexity-effective.

Routing

The routing function determines the path taken by a packet from its source to a destination. The choice of a routing function significantly affects network performance and efficiency. One of the simplest routing functions is Dimension-Order Routing, or DOR. As the name implies, packets under dimension-order routing traverse the network dimensions in order; for instance, with XY-DOR, all of the hops in the X dimension are taken before the Y dimension is navigated. DOR is attractive in that is is naturally *deadlock-free*, features low design and verification complexity, and has minimal hardware requirements.

One shortcoming of DOR is that it is unable to react to network conditions, such congestion or faults. More sophisticated routing approaches can often improve network performance by carefully spreading traffic across network links in response to network state. However, such techniques often carry undesirable side-effects. For instance, packets may be required to take non-minimal routes to the destination, which carries an energy and delay overhead that is undesirable in the on-chip setting. Similarly, a carefully orchestrated communication pattern can be disrupted by myopic routing decisions that lack a global knowledge of network state. NOCs require adaptive routing techniques that are sensitive to on-chip energy and delay constraints and that are also aware of global network conditions.

Flow Control

The flow control mechanism governs the allocation of resources to packets in a network. Buffers, switch bandwidth, and link bandwidth are the key resources in the network, although the latter two are often coupled from an allocation perspective. The granularity of allocation is important, as it determines the complexity of the allocation task and affects fairness. Packet-level allocation, as the name suggests, assigns a resource to an entire packet at once. In contrast, flit-level allocation regulates accesses to a resource one *flit*, or transmission unit, at a time. A good flow control mechanism is (a) efficient, in that it does not leave a resource idle when there is demand for it, and (b) fair, meaning that requesters of equal priority have the same probability of success in acquiring a resource. In an on-die network, a well-designed flow control mechanism can be used to reduce network area and energy costs through smart utilization of fewer resources.

Quality-of-Service

Complementary to the flow control mechanism is the QOS regulator. Whereas flow control is concerned with fair allocation of resources to contending packets and flits, QOS seeks to satisfy higher-level objectives. These include guaranteeing bandwidth, latency, or jitter bounds to individual *flows*, defined at the granularity of a thread, application, or virtual machine.

Historically, parallel machines with an interconnection network have been closely associated with high-performance computers – a domain with limited need for QOS support. In the on-chip setting, the fine-grained nature of resource sharing demands hardware QOS support for performance stability and isolation. Traditional network QOS architectures carry a high cost in terms of buffer requirements and, in many cases, arbitration complexity. What is needed for NOCs are light-weight mechanisms that can meet the applications' QOS demands without sacrificing either performance or efficiency.

Additional Terminology

node: refers to a network node with an associated router.

terminal: a discrete system resource, such as a core, cache tile, or memory controller, with a dedicated port at a network node.

virtual channel (VC): a logical buffer which is part of a physical packet memory at a router port. Virtual channels are commonly used to (a) improve performance by avoiding head-of-line blocking and (b) avert protocol deadlock by segregating packets of different priority classes [11].

2.2 Router Microarchitecture

On-chip constraints demand router designs with low latency and high area- and energy-efficiency. Figure 2.1 shows the canonical NOC virtual channel router, first described by Peh and Dally [56]. The router is input-queued and has five ports, of which four are network ports and one is an injection port. Key architectural elements of the router include the virtual channel FIFOs, route computation unit, VC allocation logic, crossbar allocation logic, and the crossbar itself. The pipeline consists of four stages: route computation (RT), VC allocation (VA), switch allocation (XA), and crossbar traversal (XB).

In this architecture, a flit enters the router through one of the network ports and is stored in a VC FIFO, which has been reserved at the upstream node. If the flit is a header, indicating the start of a new packet, it proceeds to the routing stage, which determines the output port that the packet will use. In the following cycle, the header flit attempts to acquire a virtual channel for the next hop. Upon successful VC allocation, the header flit enters the switch arbitration stage, where it competes for the output port with other flits from the router. Once crossbar passage is granted, the flit traverses the switch and enters the channel. Subsequent flits belonging to the same packet can proceed

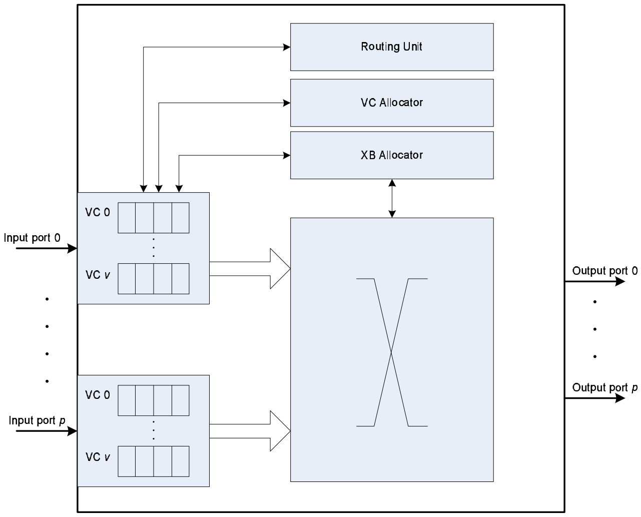


Figure 2.1: Canonical NOC router microarchitecture.

directly to switch allocation, skipping the RT and VA stages.

To reduce the impact of router pipeline delay, researchers have developed *route look-ahead*, which performs routing one hop in advance and reduces the required number of stages from four to three [24]. Another latency-hiding approach is *speculation*, which allows switch allocation to be overlapped with VC allocation [56]. If both allocation requests are granted, the latency of switch arbitration is hidden. When coupled with route look-ahead, speculation reduces the pipeline length to two cycles in the best case. Mullins et al. demonstrated that additional speculation can reduce router latency to a single cycle if the crossbar traversal is optimistically initiated in parallel with VC and switch allocation [50]. The speculation is beneficial only at low loads and misspeculation incurs a one-cycle penalty. Another drawback of the single-cycle speculative architecture is that it requires a long cycle time, which makes it unattractive for high-frequency designs.

2.3 Evaluation Metrics

2.3.1 Performance

The performance of an interconnection network is determined by two factors: throughput and latency [15]. Throughput is the maximum rate at which the network can accept and deliver the data. Latency is the time taken by a packet to traverse the network from the source to the destination and comprises two components: header latency, T_h, and serialization delay, T_s.

$$T_h = (d_r + d_w)H \tag{2.1}$$

$$T_s = \lceil L/W \rceil - 1 \tag{2.2}$$

$$T = T_h + T_s = (d_r + d_w)H + \lceil L/W \rceil - 1 \tag{2.3}$$

Equation 2.1 shows the header latency as the sum of router delay, d_r, and wire delay, d_w, at each hop, multiplied by the hop count, H. The serialization latency (Equation 2.2) is the number of cycles required by the portion of the packet following the header flit to cross a given channel, computed as the quotient of the packet length, L, and the channel width, W. The resulting expression in Equation 2.3 is known as the *zero-load* latency. In practice, contention between different packets in the network can increase the router and/or serialization delay, leading to higher packet latencies. A good network organization seeks to minimize latency and maximize throughput.

2.3.2 Area

Traditionally, the cost of interconnection networks has been dictated primarily by pin constraints of the available packaging technology. In networks-on-chip (NOCs), however, wiring complexity and die area of routers and communication channels are the main determinants of network cost.

Channel footprint is affected by the area of wires and repeaters. For wires, higher layers of the metal stack are preferred to lower-layer metal, as upper metal layers offer superior electrical characteristics due to their coarser wire dimensions, Additionally, channels routed in higher-layer metal can be run over other logic or memory, which eliminates the wires' contribution to network area. In such cases, channel footprint is dominated by the area of repeaters, which must be inserted at regular intervals to improve the delay characteristics of highly resistive on-chip wires associated with nanometer-scale process technologies. Equation 2.4 approximates the channel area as a product of the area cost per mm of wire, channel width (W), and the channel length (l), in millimeters, over the complete set of network channels (C).

$$A_{links} = A_{wire_{mm}} \cdot W \cdot \sum_{i=1}^{C} l_i \quad (2.4)$$

$$A_{routers} = (A_{fifo} + A_{crossbar} + A_{arbiters}) \cdot N \quad (2.5)$$

$$A_{NOC} = A_{links} + A_{routers} \quad (2.6)$$

Equation 2.5 calculates the routers' contribution to network area footprint as the product of the area of one router and the number of routers in the network, N. Our model accounts for three primary components in estimating router area: flit buffers, crossbar switches, and control logic. Control logic tends to have a small effect on router footprint [27, 28]. The crossbar, on the other hand, can be a significant contributor, especially in networks with wide interfaces and/or rich connectivity. Equation 2.7 approximates the area of a classical wire-dominated crossbar as a function of the number of input and output ports (P), port width, and wire pitch. For a symmetric switch, i.e., one with an equal number of input and output ports, crossbar area is quadratic in both port width and count.

$$A_{crossbar} = (P_{in} \cdot W) \cdot (P_{out} \cdot W) \cdot Pitch_{wire}^2 \quad (2.7)$$

2.3.3 Energy

NOC energy for a given workload is a function of the amount of data that traverses the network, the distance over which the data travels, and the number of router traversals involved. Here, we derive the dynamic energy for a single unit of transmission based on the energy dissipated in links and routers.

Link energy is directly proportional to the wire distance and can be approximated as the product of the energy per millimeter of wire, the channel width (W), and the sum of the wire lengths (l) over all hops (H) between the source and the destination (Equation 2.8). The summation allows for the non-uniform hop distances found in some topologies.

$$\begin{aligned} E_{link} &= E_{wire_{mm}} \cdot W \cdot \sum_{i=1}^{H} l_i && (2.8) \\ E_{router} &= \sum_{i=1}^{H} (E_{fifo} + E_{crossbar} + E_{arbiters}) && (2.9) \\ E_{NOC} &= E_{link} + E_{router} && (2.10) \end{aligned}$$

The main contributors to a router's energy footprint are the flit FIFOs, the internal switching logic, and the arbiters. FIFO energy depends on the number of virtual channels per router port as well as the depth and width of each virtual channel (VC). Switch energy is proportional to the perimeter of the crossbar. For a fixed-size transfer, switch energy scales linearly with the number of ports and their width. Finally, arbiters typically contribute a small fraction to router's energy overhead and are included in our analysis for completeness. As shown in Equation 2.9, the combined router energy must be scaled by the hop count to yield the full contribution of router energies to the total network energy.

The total energy required to deliver a single flit is simply the sum of energy values expended in routers and channels (Equation 2.10). Under this simplified model, distinct topologies that route a packet over the same Manhattan distance would incur the same link energy cost but dissimilar router energy as a result of differences in router microarchitecture and hop count.

2.3.4 Quality-of-Service

In addition to the performance, area, and energy metrics necessary for evaluating on-chip systems, QOS disciplines require dedicated metrics for measuring their fairness and performance. Key among these are *relative throughput, latency, and jitter.*

Relative throughput: The fairness criterion dictates that link bandwidth should be allotted equitably among flows, in proportion to the specified rates of service. Given the mean throughput of a set of flows with the same reserved rate, request rate and measurement interval, *relative throughput* can be measured by assessing the minimum, maximum, and standard deviation from the mean in the flow set. A system provides strong throughput fairness when each node's bandwidth consumption is close to the mean.

Latency: The end-to-end latency of a flow should be proportional to its hop count, reserved rate, and contention from other flows. In the absence of contention, the delay imposed by the QOS mechanism should be minimal. On the other hand, when two or more flows with the same specified rate converge on an output link, the QOS mechanism must ensure equal per-hop delay for the affected flows. As above, the key metrics are minimum, maximum, and standard deviation from the mean hop latency for a set of flows sharing a port.

Jitter: The variation in delay for a pair of packets in a flow is commonly called jitter. Low jitter in the face of contention provides a strong illusion of a private network for each flow, desirable for performance stability and isolation. Our metric for jitter is *packet delay variation (pdv)*, defined for IP performance measurement as "the difference in end-to-end delay between selected packets in a flow with any lost packets being ignored" [58]. The maximum *pdv* and standard deviation from the mean *pdv* within a flow, as well as across flows, are more important than the minimum observed jitter value.

2.4 Summary

On-die interconnection networks of highly-integrated chips must provide high performance through low latency and good throughput under load. At the same time, chip-level constraints necessitate network architectures that are sensitive to energy, area, and wiring costs. Traditional interconnection networks evolved without regard to these constraints, most notably the energy constraint. As a result, NOCs necessitate novel topologies, routing and flow control architectures, as well as QOS mechanisms that account for limitations of die-level systems and that also leverage their strengths.

Chapter 3

Regional Congestion Awareness

Chip-level systems are highly sensitive to network latency, energy, and delay. Since routers can significantly contribute to all three of these costs, NOCs tend to favor simple router designs with a limited number of virtual channels, low-complexity arbiters, and simple routing algorithms. While such cost-conscious architectures minimize packet latency at low loads, they often do so at the expense of diminished throughput compared to organizations with larger buffer pools and sophisticated arbitration and routing schemes.

Adaptive routing is a low-overhead technique for improving network performance without resorting to area- and energy-hungry large buffer configurations or long-latency/high-throughput arbitration schemes. Various adaptive routing schemes have been successfully used in commercial multiprocessors from IBM [2], Cray [64], and Alpha [49]. Adaptive routing boosts performance by routing packets around congested areas and flattening the distribution of traffic across the network links. In both cases, the improvement is realized through increased load balance, which smooths out non-uniformities in the original traffic pattern. In doing so, adaptive routing reduces contention for network resources, thus diminishing the need for deep packet buffers or complex arbiters. Adaptive routing requires network path diversity between source

RCA was developed in collaboration with another researcher, Paul Gratz. Portions of the text in this chapter appear in the published version of the work [29] and in his dissertation [30].

and destination nodes to facilitate load balance. The availability of network path diversity depends on the topology of the network, the traffic pattern, and whether non-minimal routes are allowed. Due to technology constraints, NOCs tend to employ low-dimensional topologies and favor minimal routing to reduce energy consumption. These features diminish path diversity and may reduce the effectiveness of adaptive routing in on-chip networks.

Existing adaptive routers used in computer interconnection networks perform output port selection for each packet based on locally-available congestion indicators. In this work, we show that reliance on local-only metrics inhibits performance due to an ignorance of global network state. Myopic routing decisions tend to upset global load balance on many workloads. In response, we introduce *Regional Congestion Awareness (RCA)*, an approach that propagates congestion information across the network in a scalable manner, improving the ability of adaptive routers to spread network load. RCA aggregates locally computed congestion metrics with those propagated from neighbors before transmitting them to upstream routers. The aggregation process naturally weighs contention information by distance from the current node so that nearby congestion influences routing more than distant congestion. We present three variants of RCA that simplify design by considering only relevant slices and regions of the network when aggregating congestion metrics.

RCA matches or exceeds the performance of conventional adaptive routing across all workloads examined, with a 16% average and 71% maximum latency reduction on SPLASH-2 benchmarks running on a 49-core CMP. The performance gains come with a negligible area overhead and no impact on a router's critical path. In addition, RCA carries a trivial energy cost, as it routes all traffic minimally and has negligible hardware requirements. These features make RCA attractive for NOCs that demand high levels of performance with low area and energy footprint.

The rest of this chapter is organized as follows. Section 3.1 summarizes relevant related work in adaptive routing for both on-chip and inter-chip interconnection networks. Section 3.2 outlines the design of our baseline router as a point of reference and describes the new elements required for capturing congestion metrics. Section 3.3 describes the RCA algorithms and variants that capture different degrees of network congestion. Section 3.4 presents performance results of RCA along with several sensitivity studies and Section 3.5 concludes.

3.1 Background and Prior Work

A paramount concern for any routing scheme, oblivious or otherwise, is its ability to balance network loads. Much research has gone into designing oblivious routing algorithms with provable worst- and average-case behavior [76, 52, 75, 65]. While these analyses typically assume a healthy network and a static load, interconnection networks frequently have non-uniform (bursty) injection rates and time-varying communication patterns [27], leading to temporary pockets of congestion known as hotspots. Schemes that have some awareness of network conditions and offer flexibility in the choice of routes often provide an advantage over oblivious approaches that are unable to adapt to the communication pattern and network state. Non-minimal adaptive routing has the potential to improve load balance beyond the limits of minimal routing [13, 67], but at the cost of greater implementation complexity and higher per-packet latency and energy. Due to the sensitivity of on-chip networks to these parameters, we restrict our evaluation to minimal routing. However, the general principles presented here could be applied to non-minimally routed networks as well.

3.1.1 Routing Policies

The routing policy determines the dynamic path taken by a given packet through an adaptively-routed network. Figure 3.1 presents a taxonomy of routing policies. Adaptive routing policies can be classified as either congestion-

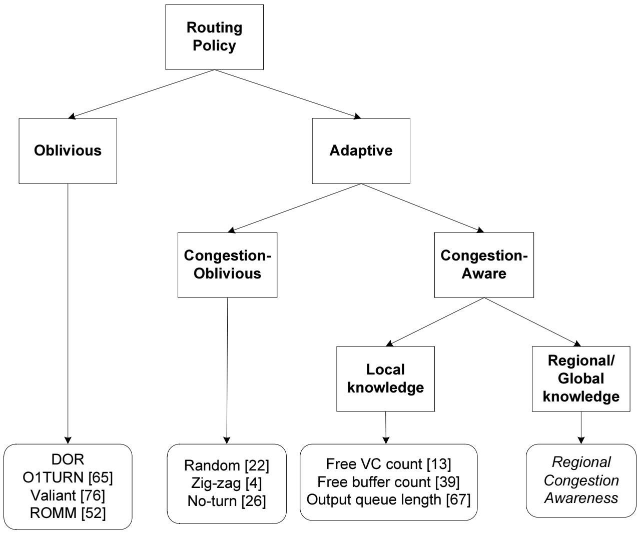


Figure 3.1: Taxonomy of routing policies with respect to congestion avoidance.

oblivious or congestion-aware, based on whether they take output link demand into account. Given a set of free and legal output ports, *random* [22] and *zigzag* [4] routing policies respectively choose an output direction randomly or based on the remaining hop count in each dimension, while *no-turn* [26] seeks to avoid unnecessary turns by following a dimension until it is either exhausted or blocked.

Congestion-oblivious routers are inherently unable to balance the load on many important traffic patterns, because they do not consider the congestion status of available ports. Congestion-aware routing policies seek to address this shortcoming. Dally and Aoki proposed to use the number of free virtual channels at an output port as a contention metric, with the routing algorithm favoring the port with the largest number of available VCs [13]. Their evaluation compared this approach to congestion-oblivious zigzag and no-turn routing and showed that congestion awareness yields lower latency and competitive throughput. More recently, Kim et al. examined buffer availability at adjacent routers as a congestion metric [39], while Singh et al. used the output queue length for the same purpose [67, 68].

Congestion-aware routing policies can be further classified based on whether they rely on purely local congestion information or take into account congestion status at other points in the network. In this context, local information is defined as information readily available at a given node, representing the status of that node or its immediate neighbors. For instance, GOAL [67] uses the queue length at each output port as its local congestion indicator during the routing phase, while GAL [68] uses the same metric for both quadrant selection and routing. A count of available virtual channels or buffers on the other end of a physical link is also local information, since it is already maintained for flow control. We define *non-local* information as originating beyond a node's immediate neighbors. To the best of our knowledge, existing evaluations of adaptively-routed interconnection networks are either congestion-oblivious or only consider local congestion indicators in their output port selection. Regional Congestion Awareness (RCA) is the first work to present a comprehensive evaluation of the utility of non-local information for improving the dynamic load-balancing properties of fully-adaptive minimally-routed networks.

3.1.2 Congestion Management

Some researchers proposed combining oblivious routing with various congestion management strategies to improve network performance. RECN dynamically allocates separate queues for flows implicated in causing congestion upstream, thus avoiding head-of-line blocking due to these flows [21]. Distributed routing balance (DRB) seeks to distribute obliviously-routed traffic by choosing one of several possible paths for each packet based on the expected latency of each route [23]. Both of these approaches depend on each packet injected into the network to follow a predetermined path – a limitation that adaptive routing does not have.

Finally, injection throttling aims improve the throughput of a network under high load by limiting injection of new packets [6, 73, 53]. Similar to congestion-aware adaptive routing, injection throttling requires knowledge of network state; however, the type of information and the way it is used is different from RCA.

3.2 Network-on-Chip Adaptive Routers

This section details the microarchitecture of a minimally-adaptive router starting from a baseline NOC router described in Section 2.2. While we restrict the discussion to a 2D mesh, this topology is not an inherent limitation of the design. We also present several router-local congestion indicators available in the target router microarchitecture.

3.2.1 Adaptive Router Microarchitecture

Given NOC's extreme sensitivity to latency, any modifications to the router microarchitecture must minimally affect router pipeline delay. Thus, adaptive routing is attractive only if it does not increase the per-hop latency. A key difference between an adaptive router and an oblivious one is that more than one legal port may be produced by the route computation unit; therefore, port selection must precede VC allocation. Two challenges complicate this process: (1) with route look-ahead, a newly arrived packet proceeds directly to VC allocation, leaving no opportunity to hide the latency of port selection prior to the VA stage; (2) VC allocation is typically on the critical path, so any major impact to the latency of this stage is undesirable.

Kim et al. proposed an elegant solution which relies on precomputation to select the preferred output direction for each packet a cycle in advance [39]. This strategy takes advantage of the fact that in a minimally routed 2D mesh, every packet travels in one of four quadrants: NE, NW, SE, and SW, with each quadrant having exactly two possible output directions, excluding the local port. The output port for each quadrant is computed every cycle for use on the following cycle based on the congestion status of each port.

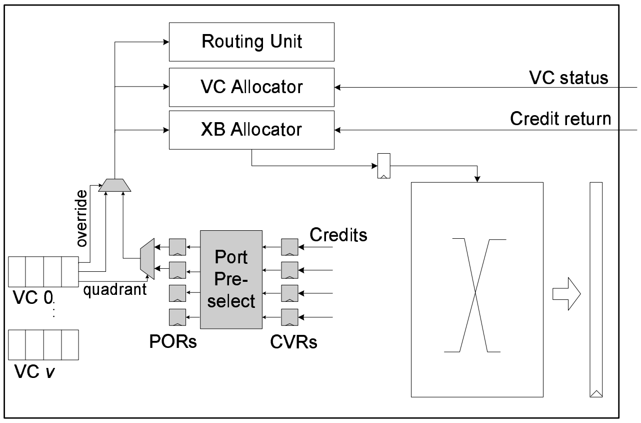


Figure 3.2: Microarchitecture of a two-stage locally-adaptive router.

Figure 3.2 shows the pipeline for a two-stage adaptive router based on Kim et al.'s design [39] with extra logic required for adaptivity shaded. The router uses free buffer count at a downstream node for congestion estimation. The counts for each port are updated every cycle and stored in the four Congestion Value Registers (CVRs). At the beginning of each cycle, Port Preselect logic reads the CVRs and computes the preferred output port for each quadrant via simple pair-wise comparisons between the registers. The port with more free buffers is the preferred output, and this result is latched in the Preferred Output Registers (PORs). A single bit in the message header is sufficient to identify the quadrant and choose the preferred output direction, as each input port in a 2-D mesh belongs to exactly two output quadrants (e.g.: West input maps to NE and SE quadrants). Once a packet reaches the final coordinate in one of the dimensions, it becomes ineligible for adaptive routing and must proceed in the remaining dimension directly to the destination. To do so, the packet must be able to override its POR value, accomplished via an override bit in the message header.

This router design can be generalized for any congestion metric whose value can be rapidly computed. For instance, using free VC count instead of buffer availability as a congestion metric requires virtually no modification to the port preselect logic. Furthermore, the low complexity of the preselect stage leaves ample room in the cycle for additional useful work. Section 3.3 explains how this slack can be exploited to integrate non-local congestion knowledge into the preselect stage to improve dynamic load-balancing properties of adaptive routers.

3.2.2 Local Contention Metrics

Any congestion metric suitable for an adaptive NOC router must correlate well with downstream congestion and be inexpensive to compute. We consider three atomic congestion metrics: free virtual channels count, available buffer count, and crossbar demand. All three metrics provide some information about downstream contention and are readily available in any reasonable virtual channel router design.

Free virtual channels (*vc*): The count of free virtual channels was first proposed as an indicator of congestion by Dally and Aoki, who noted that fewer allocated VCs implies less multiplexing on a given link [13].

Free buffers (*bf*): Kim et al. used the count of free buffers in their low-latency adaptive router [39]. Buffer count indicates the amount of backpressure that the input port at the downstream node is experiencing.

Crossbar demand (*xb*): Crossbar demand, a new metric we propose and evaluate, measures the number of *active* requesters for a given output port. Crossbar demand captures the actual amount of channel multiplexing a new packet is likely to experience. Multiple concurrent requests for an output port indicate a convergent traffic pattern, a likely bottleneck. In router architectures that employ speculation, both speculative and non-speculative switch requests are counted.

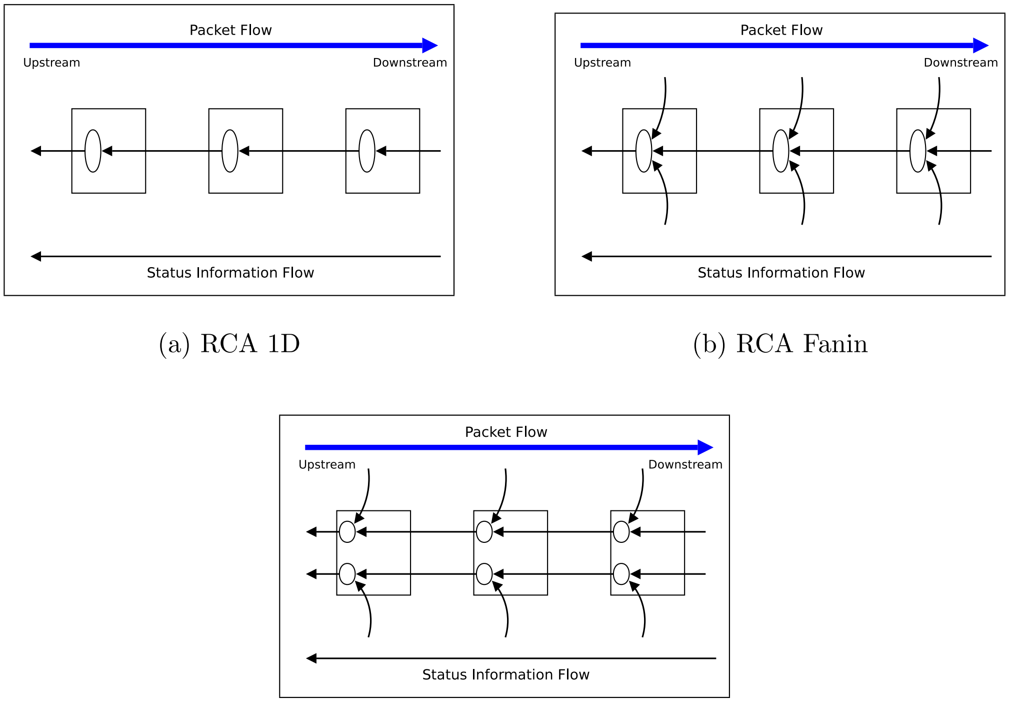
Composite metrics: Each of the atomic metrics has strengths and weaknesses. We propose simple pairings of the atomic metrics to build on their strengths and nullify their shortcomings. The three combinations of the atomic metrics are: free VCs and free buffers (*vc_bf*); free VCs and crossbar demand (*xb_vc*); and free buffers and crossbar demand (*xb_bf*).

We compared the performance of a local adaptive router using these congestion metrics across a wide range of workloads. Among non-combined metrics, *bf* and *vc* performed similarly, while *xb* performed slightly better. The combined metrics generally outperformed the non-combined, with *xb_vc* performing the best across the widest range of workloads. We examined other potential congestion metrics, but found none that performed as well as those discussed here. Interested readers are referred to Gratz's doctoral thesis for an extended analysis and evaluation of the different contention metrics [30].

3.3 Regional Congestion Awareness

Adaptive routing is useful whenever oblivious approaches lead to non-uniform link utilization. Many important workloads exhibit spatial and temporal communication patterns that can greatly benefit from adaptivity. However, certain traffic permutations, including bit-complement and uniform-random, uniformly load links in the network and enjoy a natural global balance under deterministic routing. Adaptive routing can disrupt this balance due to greedy, local decisions that lack knowledge of network state beyond the nearest neighbors. In a 2D mesh, adaptive routing tends to steer traffic toward the middle of the network, leaving the edge links underutilized and congesting the center of the mesh. Such behavior destroys the global load balance, a well-known problem shared by many existing adaptive routers.

We introduce Regional Congestion Awareness (RCA) to overcome the limitations of conventional adaptive routers, which we term *locally adaptive*. RCA is a family of scalable light-weight mechanisms for integrating congestion information from different points in the network into the port selection process. RCA does not require centralized tables, all-to-all communication, or in-band signaling that contributes to congestion. Instead, RCA uses a low-bandwidth monitoring network to propagate congestion information among adjacent routers. At each network hop, the router aggregates its local conges-



(a) RCA 1D

(b) RCA Fanin

(c) RCA Quadrant

Figure 3.3: Regional Congestion Awareness variants.

tion estimate with that of neighboring nodes. The new congestion estimate is used for port preselection and is propagated upstream. The aggregation step weighs contention information based on distance from the current node, reducing the negative effects of staleness and avoiding interference from non-minimal paths. The proposed scheme can be trivially integrated into the pipeline of a conventional locally-adaptive router, with negligible impact on area and no effect on its critical path.

3.3.1 RCA Variants

We examine three promising RCA variants with different cost-performance characteristics.

RCA 1D: This simple design aggregates and propagates congestion information along each dimension independently. RCA 1D offers excellent visibility along the axes bounding a packet's routing quadrant, but provides no direct knowledge of network status from the middle of the quadrant. Figure 3.3(a) shows how RCA 1D propagates congestion status in the West direction. While offering only limited visibility into the network, this approach has the lowest implementation complexity in the RCA design space.

RCA Fanin: The goal of RCA Fanin is to provide more information about network state than RCA 1D at minimal logic overhead. RCA Fanin provides a coarse view of regional congestion by aggregating congestion estimates along the axis of propagation with those from orthogonal directions as shown in Figure 3.3(b). While RCA Fanin encompasses significantly larger regions of the network than RCA 1D's uni-directional congestion vectors, it also introduces noise into its estimates by combining information from mutually exclusive routing quadrants.

RCA Quadrant: Depicted in Figure 3.3(c), RCA Quadrant aims to maximize the accuracy of congestion estimates by maintaining separate congestion values for each network quadrant. Doing so reduces the noise caused by combining information from mutually exclusive routing regions that exist in RCA Fanin while maximizing the coverage as compared to RCA 1D. Since each port belongs to two different quadrants, two separate congestion values must be received, updated and propagated at each network interface, incurring twice the overhead in logic and wiring complexity as either RCA 1D or RCA Fanin.

3.3.2 RCA Microarchitecture

We modify only the conventional locally-adaptive router's port preselection logic in RCA's implementation, maintaining its simplicity and low latency. As discussed in Section 3.2.1, port preselection has low logic complexity, permitting integration of additional functionality with no impact on cycle time.

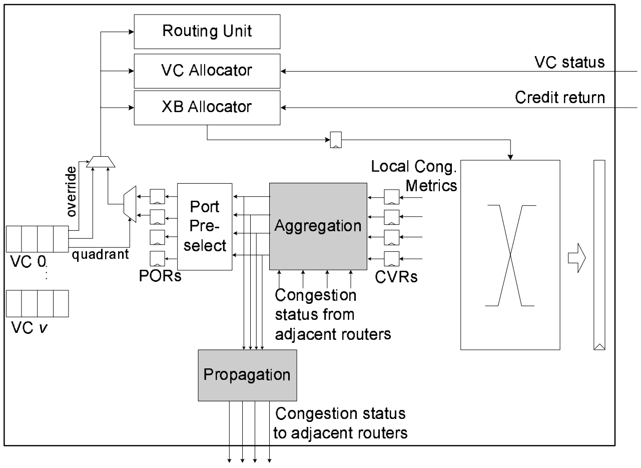


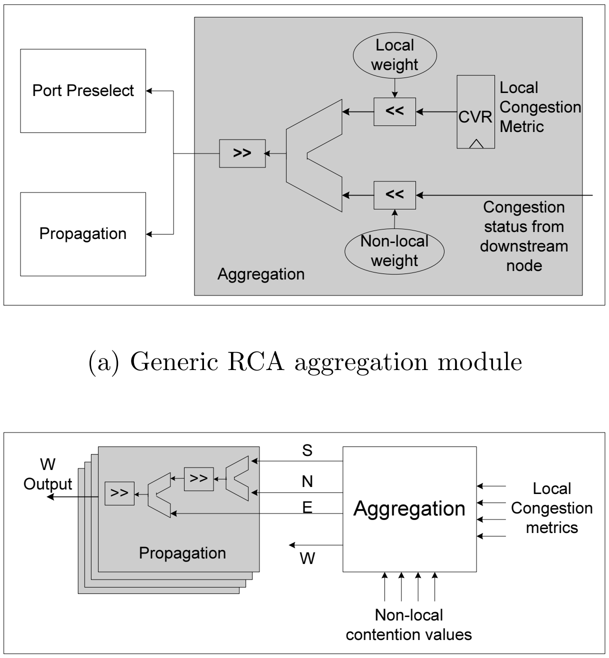
Figure 3.4: Microarchitecture of an RCA router.

Figure 3.4 shows the modifications to the 2-stage adaptive router for RCA. The two new modules we add are congestion status *Aggregation* and *Propagation*.

Aggregation

In a conventional adaptive router, local congestion estimates serve as inputs to the port preselect logic. With RCA, the port preselect logic remains unmodified, but its inputs are generated by the aggregation module, which combines local and non-local congestion estimates. An aggregation module resides at each network interface in all RCA variants, although RCA Quadrant has two such modules per port. Figure 3.5(a) shows the aggregation module in detail. Inputs to the aggregation module come from downstream routers and the local CVRs, reflecting the local congestion estimate. Aggregation logic combines the two congestion values, potentially weighting one value differently than the other, and feeds the result to the port preselect logic and the propagation module.

The exact weighting of local and non-local congestion estimates determines the dynamic behavior of the routing policy. Placing more emphasis on local congestion information moves a design toward the locally-adaptive



(a) Generic RCA aggregation module

(b) RCA Fanin propagation module

Figure 3.5: RCA microarchitectural details.

end of the spectrum. Too much weight on the non-local data increases the risk of making decisions based on remote parts of the network that may be unreachable with minimal routing. We performed a detailed empirical evaluation to determine the proper weighting of local versus non-local information, and found that the simplest assignment of weights, 50-50, is the most consistent performer across a wide set of benchmarks. Thus, aggregation is a simple matter of finding the arithmetic mean of local and non-local values, efficiently computed via an add and a right-shift. The 50-50 weight assignment makes sense, since information from nearby nodes is emphasized more than information from farther downstream in potentially unreachable network regions.

Routers on the edges of an RCA-enabled network require special treatment. In a mesh and other non-edge-symmetric topologies, edge-facing router ports are unconnected and do not receive congestion notifications. In addition, the local congestion estimates for these ports are meaningless, as no traffic ever gets forwarded through unconnected interfaces. The end result is that edge nodes may transmit misleading congestion estimates that can degrade RCA's ability to load-balance the network.

To counteract the effect of unconnected ports, we leverage congestion information from the connected ports in the orthogonal dimension, since those are the only valid network outputs for packets traveling toward the network edge. Specifically, we average congestion estimates from connected network ports in orthogonal directions. For instance, a node on the East edge of the die would transmit to the West the average of North and South congestion indicators. On the other hand, a node in the NE corner would only transmit the S congestion estimate in the West direction.

Propagation

Transmission of congestion information to adjacent nodes is performed by the propagation module, which combines congestion values computed by the router's aggregation units to reflect conditions along a given dimension, quadrant, or any other set of ports. The exact function of the propagation module differentiates the RCA variants from one another.

Figure 3.5(b) details the propagation module for RCA Fanin. At a high level, a packet arriving at a given input port can leave toward one of two quadrants. The straight-line path from a given input to an output lies in both of those quadrants, while a turn corresponds to just one of the quadrants. For instance, a packet arriving at the East input may route to either the NW or SW quadrant, so the probability of the West port being a legal output is higher than either the North or the South. The propagation module for RCA Fanin accounts for this effect by assigning 50% of the weight to the straight-line path and 25% to each of the other possible outputs. RCA Fanin's propagation logic consists of two adders and two fixed shifters. The first adder-shifter pair averages the congestion estimates from the orthogonal directions, while the second combines this average with the straight-line congestion value, creating the desired weight distribution.

The propagation module for RCA Quadrant is simpler than RCA Fanin's, as it requires only one adder and a shifter to average the aggregated congestion estimates for a given quadrant. For RCA 1D, the aggregated congestion values from each port are forwarded upstream unmodified, eliminating the need for a propagation unit.

3.3.3 Status Network Design

All RCA variants must satisfy two conflicting goals: low network bandwidth overhead and high congestion status resolution. The latter is key to early congestion detection. The averaging step in each router's aggregation unit limits the bit-width of a congestion estimate, but also leads to information loss, as one bit of congestion data is discarded per hop. With N bits of precision in a congestion estimate, a newly-aggregated value is completely discarded in N hops. To ensure that congestion information is not phased out too rapidly, the router normalizes local values by left-shifting them prior to aggregation. Normalization can be accomplished by folding the additional shift distance into the local weight adjustment in the aggregation module shown in Figure 3.5(a). The shift amount determines the minimum number of hops that a given congestion value will be "live." Empirically, we established that a shift distance of five seems to work well for our baseline 8x8 mesh. While we do not tune this parameter for any of the benchmarks, different mesh sizes and packet length distributions could likely benefit from some amount of tuning.

Assuming that a congestion metric can be summarized in three bits, plus five additional bits for normalization, both RCA 1D and RCA Fanin require eight bits per link; RCA Quadrant doubles this number to 16 bits. Given that current NOC designs feature channel widths on the order of 128 bits [27], RCA wire overhead represents just 6% for 1D and Fanin and 12%

for Quadrant. While NOCs are not generally wire limited, it may sometimes be necessary to reduce this overhead. One way to lower RCA bandwidth requirements serializes congestion updates. We experimented with a monitoring network that reduces RCA's bandwidth demand at the cost of lower update frequency. Across all of our benchmarks, results show that even bit-serial status networks (one bit per channel for RCA 1D and RCA Fanin, two bits for RCA Quadrant) do not cause noticeable performance degradations compared to a full-width RCA design. Thus, low-bandwidth RCA can be deployed in wire- or pin-constrained environments, provided traffic patterns are stable enough to tolerate reduced update frequency.

3.4 Evaluation

We evaluated the three RCA variants using both synthetic and application-based workloads, comparing them to oblivious and locally-adaptive routing techniques. We also examined RCA's sensitivity to a variety of network parameters.

3.4.1 Methodology

We use a cycle-accurate network simulator that models the two-cycle router microarchitecture from Section 3.2.1. The router model is instrumented to collect the congestion metrics proposed in Section 3.2.2 and supports all RCA variants. We measure the performance of three baseline architectures: (1) *DOR*, a dimension-ordered oblivious router; (2) *Local*, a locally adaptive router that uses the *vc* congestion metric; and (3) *Local Best*, which is an adaptive router that uses our *xb_vc* combined congestion metric. RCA 1D, RCA Fanin, and RCA Quadrant also use the *xb_vc* congestion metric. Table 3.1 details the baseline network configuration, along with the variations used in the sensitivity studies.

Feature	**Baseline**	**Variations**
Topology	8x8 2D mesh	4x4, 16x16 mesh
Routing	Minimal, fully-adaptive, reserved VC deadlock avoidance [20]	–
Router uArch	Two-stage speculative	–
Per-hop latency	2 cycles in the router, 1 cycle in the channel	–
VCs/port	8	2, 4
Flit buffers/VC	5	–
Packet length (flits)	1–6 (uniformly distributed)	1, 1–15
Traffic workload	transpose, bit-complement, uniform random, self-similar	Permutations, SPLASH-2 traces
Warmup cycles	10,000	–
Analyzed packets	100,000	200,000; whole trace

Table 3.1: Baseline network configuration and variations.

3.4.2 Workloads

We evaluate regional congestion awareness using four standard synthetic traffic patterns: *transpose*, *bit-complement*, *uniform random* and *self-similar*. These workloads provide insight into the relative strengths and weaknesses of the different congestion metrics and aggregation techniques. They represent adversarial, friendly, and nominal workloads for adaptive routing algorithms. Except for *self-similar*, all synthetic traffic patterns use a uniform random injection process. The *self-similar* traffic pattern uses a randomly generated fractional Gaussian noise distribution with a Hurst constant value of 0.8 for both the injection process and the source/destination node generation [18].

Permutation patterns, in which clusters of nodes communicate among themselves for extended intervals, are common in multiprocessor applications. We evaluate RCA on 100 randomly generated directed communication graphs at 30% injection bandwidth using the methodology similar to that of Singh and Dally [67].

Finally, we evaluate RCA on trace driven traffic generated from

SPLASH-2 benchmarks [70], representing a typical CMP scientific workload. The traces were obtained from a forty-nine node, shared memory CMP system simulator with a 7x7 2-D mesh interconnect model [44]. We configured our network simulator to match the environment in which the traces were captured.

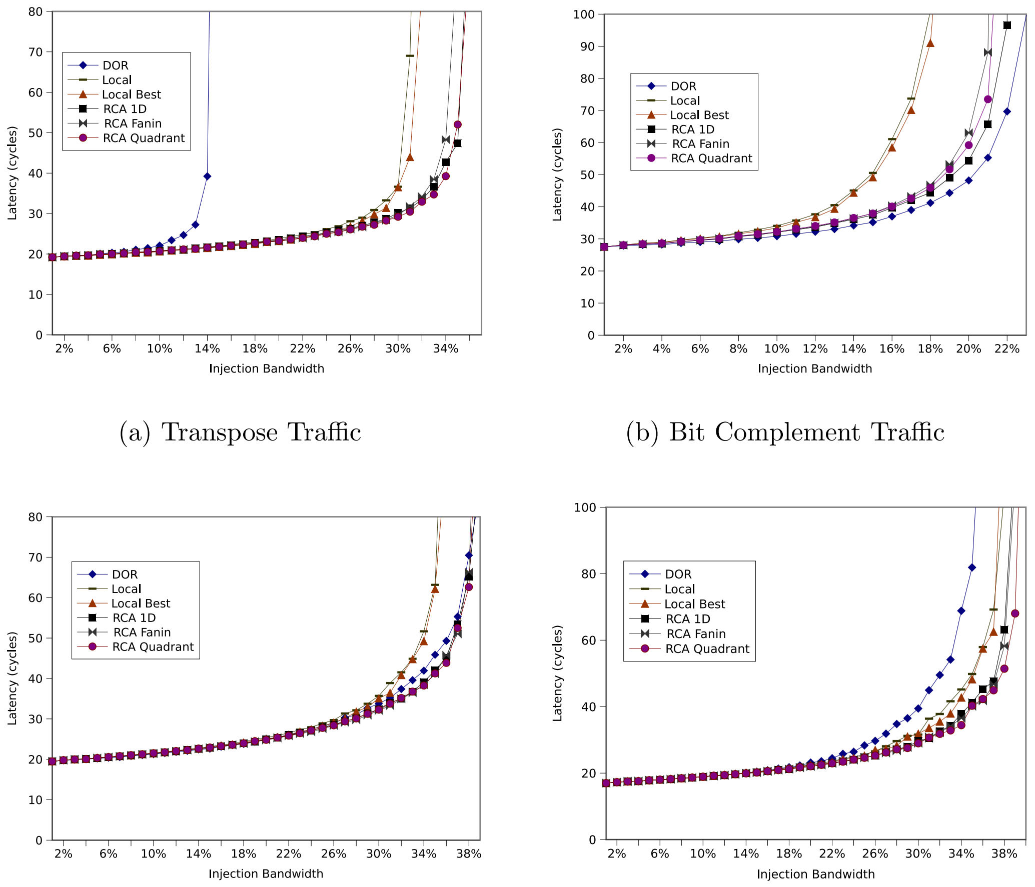
3.4.3 Evaluation of RCA Metrics

Standard Synthetic Loads

Load-latency graphs in Figure 3.6 show the latency and throughput of RCA variants, DOR, and Local routers on a set of synthetic traffic patterns. Saturation bandwidth is measured as the point at which the average packet latency is three times the zero load latency. As expected, Local provides an improvement in throughput over DOR on *transpose* and *self-similar* traffic. Load imbalances caused by DOR with these traffic patterns are egregious enough that Local metrics can detect and compensate for them. Also as expected, DOR outperforms Local on *bit-complement* and *uniform random* traffic. These traffic patterns are uniformly distributed with DOR, and Local's greedy behavior causes a significant throughput reduction. Local Best performs marginally better than Local across all traffic patterns, although the variance is under 5%.

The RCA schemes, as compared to Local, show improvement in throughput across all traffic patterns with no sacrifice in latency. The largest gain is observed on *bit-complement*, where RCA shows a 23% throughput improvement over Local, although it remains 8% shy of DOR. RCA is unable to match DOR's throughput because *bit-complement* traffic is ideally balanced under DOR routing. On all other synthetic traffic patterns, including the statistically balanced *uniform random*, RCA outperforms both DOR and Local by detecting transient load imbalances from afar and adjusting its routing decisions accordingly.

The RCA variants show very little difference across the synthetic work-



(a) Transpose Traffic

(b) Bit Complement Traffic

(c) Uniform Random Traffic

(d) Self-Similar Traffic

Figure 3.6: Performance comparison of RCA, locally adaptive, and oblivious routing.

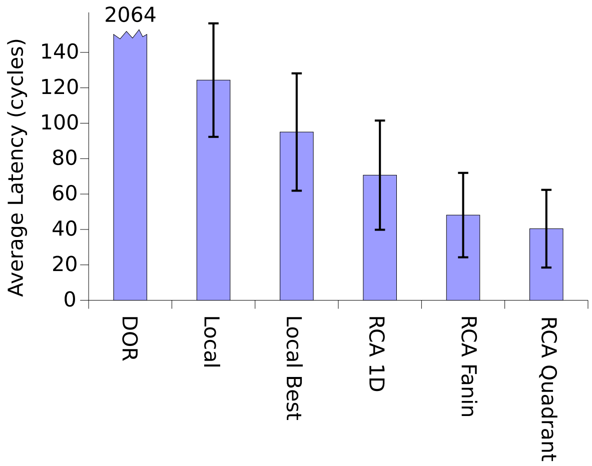


Figure 3.7: Average latency for 100 permutations of random pairs traffic at 30% injection bandwidth. Error bars show the 95% confidence interval of the mean.

loads, although typically RCA Quadrant performs best, followed closely by RCA Fanin and RCA 1D. The one exception is with *bit-complement*, in which RCA 1D outperforms both RCA Fanin and RCA Quadrant. With *bit-complement* traffic, load is a direct function of the distance from the bisection of the network. RCA 1D only considers uni-directional congestion vectors, enabling it to keep traffic flowing in lanes, similar to DOR.

Permutation Traffic

Figure 3.7 shows the packet latency averaged across 100 random permutations at 30% injection bandwidth. All adaptive approaches outperform DOR by dynamically adjusting routing decisions in response to each pattern's characteristics. RCA schemes do a better job of globally balancing the load than Local methods, yielding lower average latencies as a result. Among adaptive schemes, RCA Quadrant performs best, followed in order by RCA Fanin, RCA 1D, Local Best, and Local. Although the absolute latencies are not meaningful due to the arbitrary choice of injection bandwidth, the results show the

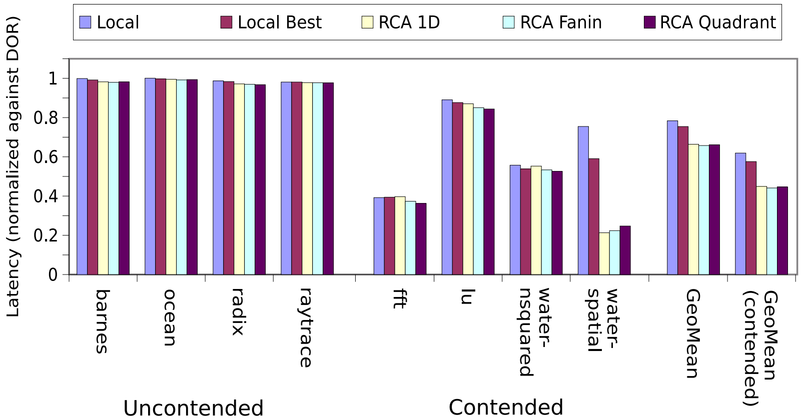


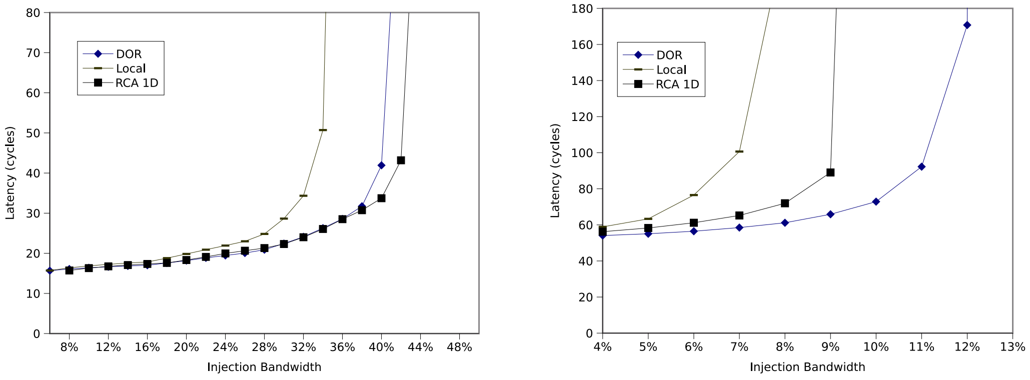
Figure 3.8: Average latency across SPLASH-2 benchmarks normalized to latency of DOR.

relative performance of the different approaches on this workload.

SPLASH-2 Benchmark Traffic

Figure 3.8 shows the average packet latency across eight SPLASH-2 benchmark traces,normalized to DOR, grouped into uncontended and contended categories. In uncontended benchmarks (*barnes*, *ocean*, *radix*, and *raytrace*) contention forms less than 15% of the total packet latency. Contention is the cause of significant packet latency in *fft*, *lu*, *water-nsquared*, and *water-spatial*; thus adaptive routing has an opportunity to improve performance. The final two clusters of bars in Figure 3.8 show the geometric mean across all benchmarks and across the contended benchmarks.

Although RCA variants provide equal or lower latency than Local schemes, RCA shows the greatest benefit on *water-spatial*, with a 71% reduction in latency. This application's traffic contains a single, localized hotspot which RCA detects, allowing it to route packets around it before they encounter congestion. On average, RCA provides a latency reduction of 16% across all benchmarks, and 27% across contended benchmarks versus Local. All three RCA variants show similar performance on these benchmarks.



(a) 4x4 Mesh

(b) 16x16 Mesh

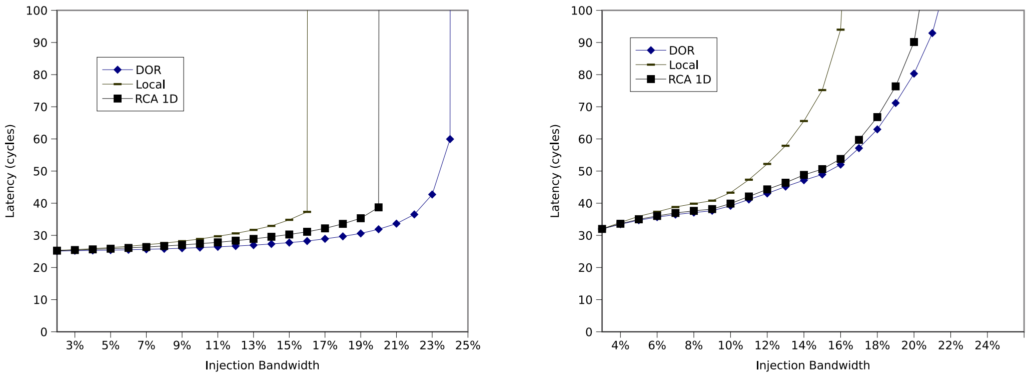
Figure 3.9: Performance in 4x4 and 16x16 meshes on *bit-complement* traffic.

3.4.4 Sensitivity to Network Design Point

Individual network implementations are likely to vary from the baseline designs of the previous section, depending on the needs of the system. Here we present variations that provide insight into the performance of RCA metrics in different environments. We do not tune RCA to better accomodate various network configurations. Results are shown only for the *bit-complement* traffic pattern, an adversarial workload for adaptive routers which offers insight into RCA's relative performance against both Local and DOR. The graphs in this subsection may be compared against the baseline configuration with *bit-complement* traffic in Figure 3.6(b). In our experiments with this traffic pattern, the variance between RCA schemes is under 5%, so only RCA 1D is shown in subsequent figures.

Network Dimension

On-chip networks are likely to exhibit a great deal of variation in size from design to design. Figure 3.9 shows load-latency graphs for two different network sizes: 4x4 and 16x16. The results for the 4x4 mesh, in Figure 3.9(a), show that RCA performs very well, achieving 25% better throughput than Local and



(a) Short Packets (1 flit)

(b) Long Packets (1-15 flits)

Figure 3.10: Performance with short and long packets on *bit-complement* traffic.

slightly exceeding that of DOR. On smaller networks, RCA provides excellent visibility into the congestion state of the network, allowing it to capitalize on the transient hotspots caused by the random injection process.

Figure 3.9(b) shows the results for the 16x16 network. On this traffic pattern, adaptive approaches do not perform as well versus DOR. The performance loss of RCA relative to DOR is caused by a reduced visibility horizon and increased noise in congestion estimates due to a large network diameter. Network size has a stronger effect on Local than RCA, extending RCA's performance advantage to approximately 25%.

Packet Length

Figure 3.10 shows the load-latency graphs for very short (1 flit) and longer (1-15 flits) packets. Short packets, shown in Figure 3.10(a), represent an NOC where many small values are transfered, such as in a scalar operand network [72]. Compared to the baseline *bit-complement* results in Figure 3.6(b), the gap between the adaptive approaches and DOR is somewhat larger. RCA continues to perform well relative to Local, showing a 15% improvement in

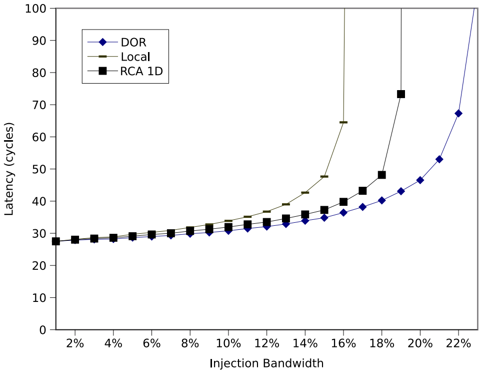


Figure 3.11: Performance with four virtual channels on *bit-complement* traffic.

its saturation bandwidth. Single-flit packets cause highly transient network congestion which is difficult for adaptive routing to exploit, increasing the gap between all adaptive routers and DOR.

The larger distribution of packet lengths (from 1 to 15 flits) in the experiment shown in Figure 3.10(b) are more representative of packet sizes found in networks for memory traffic. The average packet latencies for both the adaptive and DOR routers are significantly higher for long packets than for short, even discounting the latency due to packet length. The increased latency is a known effect of wormhole routing with long packets, where imbalances in resource utilization arise because packets hold resources over multiple routers. RCA capitalizes on this phenomenon to provide an accurate picture of network utilization and improve routing decisions, almost matching the performance of DOR.

Virtual Channel Count

Figure 3.11 shows a load-latency graph for a modified baseline configuration with the virtual channel count reduced to four. RCA continues to perform significantly better than Local, delivering an improvement of 18% in throughput, although the performance gap is reduced. Fewer virtual channels, and by extension fewer flit buffers, reduce the resolution of various contention metrics and cause diminished performance in RCA. Another issue is the imbalance in virtual channel utilization caused by the presence of the escape VCs in the Y direction. The escape VCs are reserved for packets on the last leg of their network traversal and cannot otherwise be used. Our contention metrics do not account for the special status of these VCs, and end up providing a misleading picture of resource availability. The attenuating effect of reserved VC's on the accuracy of congestion estimates is amplified as the number of VCs is reduced, a trend confirmed with experiments simulating two VCs per physical channel.

3.4.5 Discussion

Across a wide range of synthetic and trace-based workloads, the RCA variants match or outperform existing locally-adaptive routers. RCA performs particularly well when the traffic pattern is highly asymmetric as in the *water-spatial* SPLASH-2 benchmark. RCA also performs well on workloads where greedy, local decisions can hurt global load balance, such as *bit-complement* traffic.

RCA's impact is reduced when the network diameter is large, or when congestion is highly transient. A large network diameter reduces the effectiveness of RCA designs because, with a 50-50 weighting of local and propagated contention metrics, small fluctuations in local metrics can outweigh strong distant trends. To improve performance of RCA in large meshes, one might consider tuning local versus non-local weights, increasing RCA bit-width for greater visibility, or using concentration to reduce network diameter [5]. Highly transient traffic patterns also complicate adaptive routing's ability to get an accurate picture of network state, leading to some performance loss. This affects any adaptive router design, although RCA reacts more quickly to network state transitions than Local.

Among RCA variants, RCA Quadrant generally performs the best, although the simplest RCA variant, RCA 1D, performs best on *bit-complement* and *water-spatial.* RCA 1D shows that less information can sometimes provide a clearer picture of network state by reducing the noise in congestion estimates. RCA Fanin performance typically lies between that of RCA Quadrant and RCA 1D, reflecting the attenuating effects of noise caused by aggregation of status information from mutually exclusive routing quadrants.

With a locally-adaptive router as a baseline, the addition of RCA carries a negligible area and energy overhead. Principal hardware additions needed to support RCA consist of a small number of adders, shifters, and registers, all of which have a narrow bit width. In fact, our evaluation suggests that RCA can actually *improve* the area- and energy-efficiency of an adaptive router through lower buffer requirements. Across a number of simulated workloads, a 4-VC RCA design is able to match or exceed the performance of an 8-VC Local router, thus making RCA an attractive option for cost-constrained on-chip networks. RCA may also reduce chip-level energy consumption by reducing the average latency of packets traversing the NOC, in turn reducing the task execution time and the energy consumed per task.

3.5 Summary

Effective routing algorithms make best use of the link bandwidth and spread traffic as necessary to balance the load. Ideal adaptive routing algorithms would accurately predict future congestion and route each message to minimize the contention. Since such an approach is unrealistic, most adaptive routing algorithms employ simple local congestion metrics in each router to determine where to next send any given message.

This chapter introduces Regional Congestion Awareness (RCA) which exploits non-local and local congestion information. A light-weight monitoring network aggregates and transmits metrics of congestion throughout the network so that each router has a better picture of network hotspots. We present three variants of RCA that differ in how routers contribute to the estimate of global contention.

An evaluation of RCA reveals that it reduces network latency and improves throughput under load as compared to traditional locally-adaptive routing approaching. Because RCA routes all traffic over minimal paths, it does not incur additional energy or delay costs of non-minimal routing policies. Additionally, RCA has very modest hardware requirements that result in trivial area and energy overheads. As a result, RCA can significantly boost network performance without sacrificing the efficiency demanded by on-chip interconnects.

While we have focused on mesh-based topologies in the context of NOCs, our approach is applicable to off-chip networks and a variety of topologies. For example, as noted in Section 1.1.2, tori are a popular choice for high-performance computer interconnects. A valuable feature of torus networks is that they are amenable to simple non-minimal adaptive routing algorithms. Such routing algorithms often include a phase in which packets are routed minimally within a given quadrant of the network, a phase to which RCA can be adapted directly. We also expect that RCA can be extended to non-minimal adaptive routing by simultaneously considering non-local contention and hop-count toward the destination in each dimension.

Chapter 4

Express Cube Topologies for On-Chip Interconnection Networks

As discussed in Chapters 1 and 2, on-chip networks are characterized by short communication distances, lack of pin constraints, and high sensitivity to latency, area, and energy costs. These features argue for NOC topologies that are amenable to implementation in planar substrates, make effective use of available wire resources, avoid non-minimal channel spans, and feature low-complexity routers to maximize efficiency and performance.

Most existing networks-on-chip (NOCs) are based on rings [57] or two-dimensional meshes [80, 77, 60, 78]. While these topologies have modest design complexity and map well to planar silicon substrates, their limited connectivity presents serious scalability challenges as the number of interconnected components on a chip grows into hundreds or thousands. Poor connectivity results in a high average hop count on many workloads, which carries a significant energy and delay overhead due to the costly router traversals. Topologies with

Parts of this chapter appear in the Proceedings of International Symposium on High-Performance Computer Architecture, 2009 [31]. The evaluation presented in this thesis has been refined from the published version to reflect improvements to the simulation infrastructure and has been updated with area and energy models for 32 nm technology.

richer connectivity have the potential to improve NOC performance and energy efficiency. However, richly-connected network organizations necessitate higher router complexity and may require significant channel resources. Topological limitations of two-dimensional substrates further restrict the space of implementable networks.

To address the scalability bottlenecks of existing topologies, we introduce Multidrop Express Channels (MECS) – a new network organization based on express cubes [10] that is specifically designed to fit the unique advantages and constraints of NOCs. MECS utilizes a point-to-multipoint communication fabric that provides a high degree of connectivity in a bandwidth-efficient manner. We use an analytical model to understand how MECS and several previously proposed topologies scale when the network size is increased from 64 to 256 terminals. As the network is scaled, MECS maintains a low network diameter and requires only a linear increase in bisection bandwidth to keep the channel width constant. An evaluation of MECS on a subset of the PARSEC benchmark suite in a 64-terminal system shows that MECS enjoys a latency advantage exceeding 9% over other topologies. Scalability studies with synthetic benchmarks show that the latency benefit of MECS increases to over 18% at low loads in a 256-terminal configuration. A detailed study of area- and energy-efficiency of various topologies reveals that MECS has a modest area footprint and near-best energy profile.

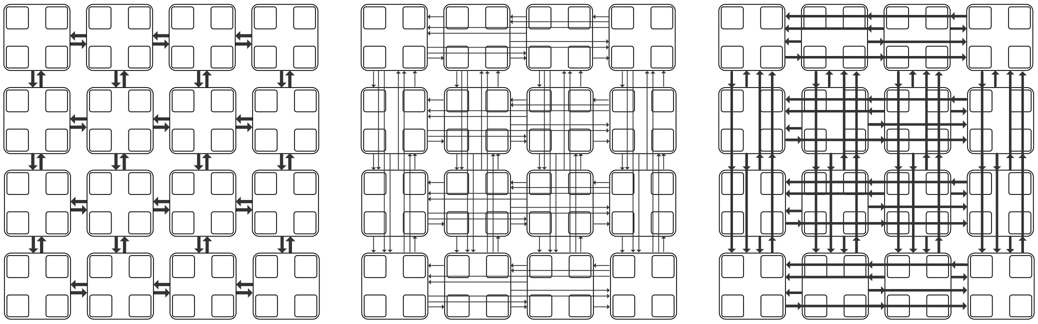
To better understand the space of on-chip interconnects, we propose Generalized Express Cubes (GEC) – a framework that extends k-ary n-cubes with concentration and express channels – and demonstrate how various topologies, including MECS, can be expressed in it. We evaluate several GEC-expressible networks that differ in channel count, connectivity and bandwidth. Our findings show that in wire-rich substrates, completely replicating the networks while holding the bisection bandwidth constant can significantly improve network throughput at a modest delay penalty at low loads. In addition, replication can reduce router area and power by decreasing the crossbar complexity.

The rest of this chapter is organized as follows. Section 4.1 surveys the relevant prior art in on-chip interconnect topologies. Section 4.2 introduces Multidrop Express Channels as a cost-effective and scalable fabric for NOCs and analytically compares MECS to previously proposed topologies. Section 4.3 discusses variations on MECS aimed at cost/performance improvements and generalize these to reveal a spectrum of on-chip interconnect topologies. Section 4.4 presents our experimental methodology, followed by the results in Section 4.5, and a sensitivity analysis of MECS to network parameters in Section 4.6. Section 4.7 summarizes the contributions of this work.

4.1 Background

As noted in earlier chapters, most NOC designs that have been manufactured to date are based on ring and mesh topologies [57, 74]. Despite their low-cost and low-complexity, simple rings appear to be the least scalable option since the hop count – and thus, latency and energy – grows linearly with the number of interconnected elements in a ring. Meshes fare better since the network diameter is proportional to the perimeter of the mesh and scales in the square root of the node count. However, a large fraction of the latency and energy in a mesh is due to the router at each hop, thus motivating the need for a more scalable topology.

Researchers have tried to address the problem of poor NOC scalability. One solution proposed by Balfour and Dally is *concentration*, a technique which reduces the total number of network nodes by sharing each network interface among multiple terminals via a crossbar switch [5]. A mesh network employing 4-way concentration, shown in Figure 4.1(a), leads to a 4x reduction in the effective node count at the cost of higher router complexity. Compared to the original network, a concentrated mesh has a smaller diameter and, potentially, a diminished area footprint. While concentration is an enabling element in the design of scalable networks, it is not sufficient by itself due to the poor scalability of the crossbar switch fabric. As explained



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

Figure 4.1: Concentrated Mesh, Flattened Butterfly and MECS topologies in a 64-terminal network.

in Section 2.3, crossbar area increases quadratically with the number of interconnected elements, a feature which makes many-ported crossbar switches expensive.

Das et al. explored an alternative router switch fabric design based on a combination of a bus and a crossbar switch [16]. In the proposed configuration, the crossbar switches only the global network traffic, while the terminals use the bus to communicate among themselves and access the global network via a shared crossbar interface. The resulting organization reduces crossbar complexity but incurs a bandwidth and delay penalty. Since both crossbar- and bus-based fabrics are difficult to scale beyond a small number of nodes, a combination of the two is similarly limited in its scalability potential, a point acknowledged by the authors of this work [16].

Researchers have also attempted to improve NOC scalability through a low-diameter *butterfly* architecture used in high-performance off-chip interconnects. The proposed *flattened butterfly* topology maps a richly connected butterfly network onto a two-dimensional substrate using a two-level hierarchical organization [40]. In the 64-terminal network, shown in Figure 4.1(b), the first level employs 4-way concentration to connect the processing elements, while the second level uses dedicated links to fully connect each of the four concentrated nodes in each dimension.

The flattened butterfly is a significant improvement over the concentrated mesh in that it reduces the maximum number of hops to two, minimizing the overall impact of router delay at the cost of increased network complexity. The flattened butterfly also makes better use of the on-chip wire bandwidth by spreading it over multiple physical channels. Unfortunately, the topology is not truly scalable, as the physical channel count in each dimension grows quadratically with the number of nodes in the dimension. In addition, the use of a large number of dedicated point-to-point links and the resulting high degree of wire partitioning leads to low channel utilization, even at high injection rates. Although channel utilization can be improved through the use of non-minimal paths, this approach requires a more complex routing and buffer reservation scheme and increases network power consumption [40].

4.2 Multidrop Express Channels

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

4.2.1 Overview

Multidrop Express Channels (MECS) are based on point-to-multipoint unidirectional links that connect a given source node with multiple destinations in a given row or column. The high degree of connectivity afforded by each MECS channel enables richly connected topologies with fewer bisection channels and higher per-channel bandwidth than point-to-point networks with similar connectivity. Figure 4.1(c) shows a 64-terminal MECS network with 4-way concentration that reduces the number of bisection channels of the comparable flattened butterfly by a factor of two. The key characteristics of MECS are as follows:

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

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

4.2.2 Microarchitecture

Router Organization

Figure 4.2 shows a high-level view of a MECS router with $2(k-1)$ network inputs and four network outputs. All inputs from a given direction share a single crossbar port for a total of four network interfaces. This organization keeps the crossbar complexity low, minimizing area and delay. The number of local switch ports depends on the degree of concentration and the degree of

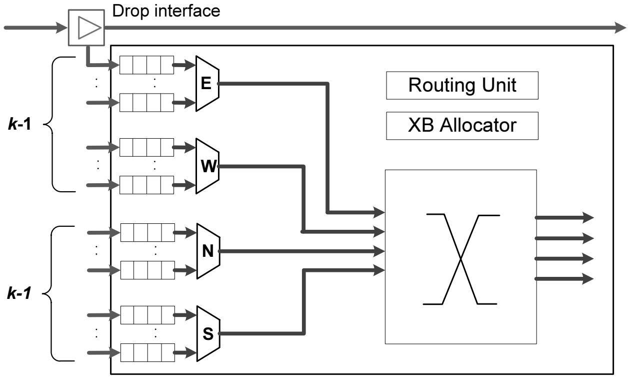


Figure 4.2: MECS router microarchitecture.

muxing into and out of the switch. While local switch ports are not shown in the diagram to reduce clutter, a 4-way concentrated router with a dedicated crossbar port per terminal would require four switch interfaces in addition to the network interfaces. Arbitration complexity in a MECS router is comparable to that in a concentrated mesh router with an equivalent $p_{in} \cdot v$ product.

Figure 4.3 shows a more detailed view of a MECS router in a 4x4 network. There are three input ports from each direction, some of which may be unconnected depending on the router's location in the network. Two-level distributed allocators first arbitrate among the input ports from a given direction and then allocate downstream virtual channels and switch bandwidth to the winners of the first stage.

An important aspect of the router microarchitecture in Figure 4.3 is the organization of the input ports. In this design, all of the inputs from the same direction are *stacked* next to each other in a column-wise fashion. A uni-directional multi-driver bus connects the ports in each stack to the crossbar's input interface. There is a total of four port stacks, one per cardinal direction, and each stack has a dedicated switch interface. Wires for the incoming network channels and the switch bus are routed directly over the ports in a stack. Bus access is granted to the winner of the switch arbitration stage. Since only one network port from each direction can access the switch in

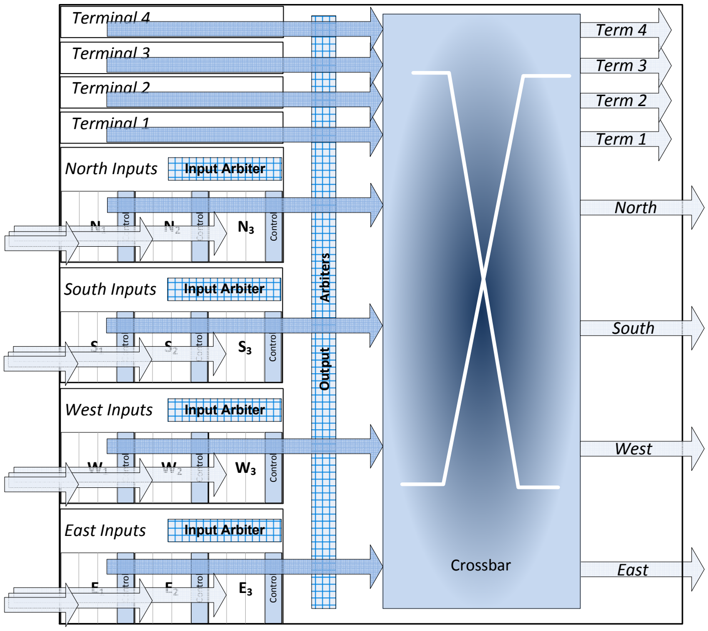


Figure 4.3: Detailed MECS router microarchitecture.

a given cycle, the granted input gets to use the switch and the bus. Compared to the organization in Figure 4.2, the stacked layout reduces wire lengths, eliminates the need for a mux at every switch interface, and yields a more compact router layout. In general, a number of variations and optimizations to this design is possible, such as varying the number of virtual channels or adding extra crossbar ports. We explore several options in Section 4.6.

Drop Interface

MECS channels integrate drop interfaces at each router along the span of the channel, as shown in Figure 4.2. The drop interfaces are responsible for steering the flit into the router input port if the associated node is the packet's destination. Otherwise, they act as repeaters and propagate the flit to the next router along the channel.

We design the drop interfaces as repeaters augmented with light-weight

		CMesh			Flattened Butterfly			MECS		
Network	Network size		64	256		64	256		64	256
	Network radix, *k*		4	8		4	8		4	8
	Concentration, *c*		4	4		4	4		4	4
	Network diameter	$2(k-1)$	6	14	2	2	2	2	2	2
BW	Bisection BW, B_B		4,608	18,432		4,608	18,432		4,608	18,432
	Row/col channels	2	2	2	$k^2/2$	8	32	k	4	8
	BW/channel, *w*		576	1152		144	72		288	288
Router	Input ports, p_{in}	4	4	4	$2(k-1)$	6	14	$2(k-1)$	6	14
	Output ports, p_{out}	4	4	4	$2(k-1)$	6	14	4	4	4
	Crossbar complexity $((p_{out}+c)\cdot w)^2$		21.2*e*6	84.9*e*6		2.1*e*6	1.7*e*6		5.3*e*6	5.3*e*6
	VCs per p_{in}, α		8	8		1	1		1	1
	VC depth, β		5	5		10	15		10	15
	Buffer size, *bits* $p_{in}\cdot w\cdot\alpha\cdot\beta$		92,160	184,320		8,640	15,120		17,280	60,480
Perf, Energy	Avg hops/packet, *H* (random traffic)		2.7	5.3		1.7	1.8		1.7	1.8
	Avg latency/pkt, *cycles*		13.2	24.2		11.7	17.6		10.7	14.1
	Avg energy/pkt, *nJ* $E_{links}+E_{routers}$		0.26	0.31		0.21	0.22		0.21	0.23

Table 4.1: Comparison of Concentrated Mesh (Cmesh), Flattened Butterfly, and MECS topologies.

logic that examines the flit's destination field and determines whether the flit should be repeated or steered into the local node. To conserve channel power, flits steered into a router are not propagated downstream.

4.2.3 Analysis

Table 4.1 compares characteristics of the concentrated mesh (Cmesh), flattened butterfly, and MECS topologies using several metrics. For each topology, the first column (in gray) provides analytical expressions for computing parameter values. The second column for each topology quantifies the parameters for a 4-ary mesh with 4-way concentration (64 terminals), and the third column repeats the analysis for a 8-ary mesh also with 4-way concentration (256 terminals). A few trends are worth highlighting:

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

Bandwidth: We keep the bisection bandwidth constant across all networks. To derive the per-channel bandwidth, we divide the total bisection bandwidth by the number of bisection channels in each topology. Thus, topologies that have fewer channels feature wider links, while the opposite is true in networks with a large number of channels. Going from 64 to 256 terminals, we grow the bisection bandwidth to match the doubling in network radix.

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

Crossbar: We approximate crossbar size as $(switch\ ports \cdot \frac{BW}{port})^2$. This value is highest for Cmesh and lowest for the flattened butterfly. While crossbars in the flattened butterfly have significantly more ports than those in other topologies, their area is small because crossbar bandwidth in the flattened butterfly is only a fraction of the bisection bandwidth. MECS topologies have considerably higher per-channel bandwidth than the flattened butterfly, but since the number of crossbar ports in MECS routers is low, the MECS crossbar area is only modestly higher than that of the flattened butterfly and significantly lower than that of Cmesh. Both MECS and the flattened butterfly are amenable to crossbar optimizations that can further reduce complexity by eliminating unnecessary switch ports from the routers.

Buffering: To estimate the buffer requirements, we assume that the Cmesh requires multiple VCs per port to avoid head-of-line blocking [5]. Both the flattened butterfly and MECS topologies can tolerate one VC per packet class per port, mitigating the adverse effects of head-of-line blocking through multiple ports. This organization also keeps arbitration complexity manageable in NOC high-radix routers. The depth of each VC is set to cover the maximum round-trip credit return latency. In practice, the depth could be adjusted for each router port based on the router's location and channel length.

This type of customization could yield a reduction in buffer area and power at the cost of higher design complexity. To keep the analysis and evaluation simple, we do not exploit such an optimization. Thus, both the flattened butterfly and MECS require greater buffer depth than the Cmesh to cover the wire delays associated with longer channels.

With these assumptions, the Cmesh requires the most buffer space, followed by MECS and the flattened butterfly. The flattened butterfly has relatively low buffer requirements because only a fraction of the bisection bandwidth reaches each router due to the high degree of channel partitioning. As the network is scaled to a larger number of nodes, the per-channel bandwidth shrinks as the port count grows, leading to a slower growth in buffer requirements. In contrast, the amount of per-channel bandwidth stays flat in MECS. As a result, in the larger network, each MECS router requires more buffering than in the smaller system to accommodate the increase in the number of input ports.

Energy and Latency: To estimate the energy requirements and performance potential of each topology, we assume a uniform random packet distribution and employ energy models for wires and routers described in Section 4.4. The bottom rows of Table 4.1 show the expected latency and energy of a single packet in an unloaded network, based on the average number of hops in each topology.

With 64 terminals, the Cmesh experiences a higher transmission latency than the low-diameter topologies due to its higher hop count. MECS, on the other other hand, observes the lowest latency, as it enjoys the same low hop count of the flattened butterfly and a decreased serialization cost due to wider channels. Scaled to 256 terminals, the latency for Cmesh nearly doubles due to the larger network diameter, while latencies for the flattened butterfly and MECS increase by 50% and 32%, respectively. The gap in per-packet latency between MECS and flattened butterfly widens to 3.5 cycles as a result of increased serialization in the flattened butterfly topology, giving MECS a 20% latency advantage.

Energy results track our analytical estimates of complexity and show that the Cmesh is the least efficient topology, consuming nearly 24% more energy than MECS and the flattened butterfly in the 64-terminal network. The latter is the most energy-frugal topology of the three, a direct result of small hop count and low crossbar complexity. The efficiency gap between the Cmesh and the low-radix networks increases to 35-40% in the 256-terminal configuration. In the larger network, MECS is roughly 5% less efficient than the flattened butterfly due to wider channels that dissipate more energy in the switch fabric.

4.2.4 Multicast and Broadcast

Parallel computing systems often provide hardware support for collective operations such as broadcast and multicast [63, 25]. MECS can easily be augmented to support these collective operations with little additional cost because of the multipoint connectivity. A full broadcast can be implemented in two network hops by first delivering the payload to all of the network nodes in a single dimension connected to the MECS channel. Each of those nodes then broadcasts the payload to all of its siblings in the second dimension. Further discussion of these features is beyond the scope of this work; here we focus on traditional NOC workloads that require point-to-point communication.

4.2.5 Comparison to Prior Work

MECS bear some resemblance to conventional multi-drop (broadcast) buses; however, a bus is an all-to-all medium, whereas MECS is a one-to-many topology. The YARC router [62] employed an 8x8 grid of switches to implement a radix-64 router. Each switch was connected to other switches in a given row via a dedicated bus, making it a one-to-many configuration similar to MECS. In each column, point-to-point channels connected each switch to others, analogous to the flattened butterfly configuration. The difference between YARC and MECS is our use of uni-directional one-to-many channels in both dimensions with intelligent repeaters for conserving power. This gives MECS desirable scalability properties in terms of performance and energy efficiency.

Kim et al. [40] proposed to extend the flattened butterfly with non-minimal routing through the use of bypass links, which provide additional exit points in an otherwise point-to-point channel. This bypassing is similar in nature to the multi-drop capability in MECS. However, its use in the flattened butterfly network requires a complex reservation protocol as input ports are shared between regular and bypassed channels. MECS do not need special routing, have dedicated input ports, and require significantly fewer channels.

Finally, Express Virtual Channels [44] attempt to reduce the latency and energy overhead of routers in a mesh topology through an aggressive flow control mechanism. MECS is a topology which also aims to eliminate the impact of intermediate routers and has a broader objective of making efficient use of the available on-chip wire budget.

4.3 Generalized Express Cubes

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

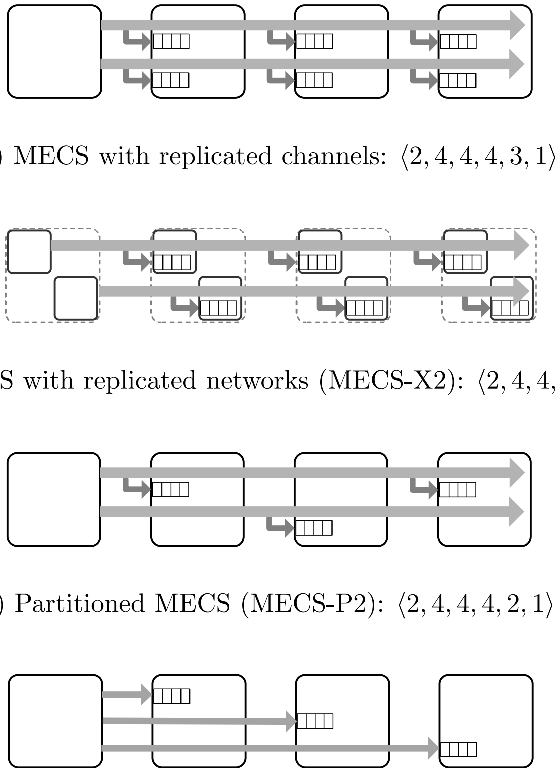
Building on the k-ary n-cube model of connectivity, we define the six-tuple $\langle n, k, c, o, d, x \rangle$ as:
n - network dimensionality
k - network radix (nodes/dimension)
c - concentration factor (1 = none)
o - router radix (output channels/dimension in each node)

d - channel radix (sinks per channel)
x - number of networks

The router radix, o, specifies the number of output channels per network dimension per (concentrated) node, equalling two in a mesh (one in each direction) and three in the flattened butterfly of Figure 4.1(b). The channel radix, d, specifies the maximum number of sink nodes per channel. A value of one corresponds to point-to-point networks; greater values define MECS channels. Finally, replicated topologies, which allow bandwidth to be distributed among several networks, can be expressed via the x parameter. Using this taxonomy, the six-tuple for the 64-terminal MECS network from Figure 4.1(c) is $\langle 2, 4, 4, 2, 3, 1 \rangle$, indicating that the baseline topology is a single-network 4-ary 2-cube employing 4-way concentration, with radix-4 routers and up to three nodes accessible via each channel. In general, we note that this taxonomy is not sufficient to specify the exact connectivity for a large set of networks encompassed by the GEC model. Here, our intent is to focus on a small set of regular topologies attractive for on-chip implementation with sufficiently diverse characteristics.

Figure 4.4 shows several possible topologies (one-dimensional slices) that can be specified with the GEC model, with MECS (from Figure 4.1(c)) at one end of the spectrum and the flattened butterfly (Figure 4.4(d)) at the other. In wire-rich NOCs, channel bandwidth may be wasted when link width exceeds the size of a frequently-occurring but short packet type, such as a read request or a coherence transaction. Wide channel topologies, like CMesh and MECS, are vulnerable to this effect while narrower channel topologies, such as the flattened butterfly, are less susceptible. Partitioning the bandwidth across multiple channels can reduce this type of wasted bandwith.

One means of partitioning, shown in the $\langle 2, 4, 4, 4, 3, 1 \rangle$ network of Figure 4.4(a), divides each baseline MECS channel into two, each with one-half of the bandwidth. This configuration can improve bandwidth utilization and reduce head-of-line blocking for short packets by doubling the router radix, o. Latency for long packets, however, might suffer as a result of increased



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

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

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

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

Figure 4.4: MECS variants for cost-performance trade-off. Only one dimension is shown for simplicity.

serialization delay. Another potential downside is an increase in arbitration complexity due to the doubling in the number of ports.

An alternative approach replicates the networks, increasing x, such that each network has full connectivity of the original but with a fraction of the bandwidth. The resuling $\langle 2, 4, 4, 2, 3, 2\rangle$ topology is shown in Figure 4.4(b). An advantage of such a design is that it does not increase the router radix and reduces the combined crossbar area. While replicated topologies have been proposed in the past to exploit greater bandwidth with a given router design [5, 28], our work shows that replication in wire-rich substrates can yield significant throughput gains and energy savings for a given bisection bandwidth.

A third option that enables a more aggressive area reduction in a MECS topology at a cost of reduced performance partitions each multidrop channel into two (or more), interleaving the destination nodes among the resulting links. The $\langle 2,4,4,4,2,1\rangle$ network of Figure 4.4(c) increases the router radix o and decreases the channel radix d. This *partitioned* MECS, or MECS-P2, topology has reduced network buffer requirements proportional to the partitioning factor and can decrease router crossbar complexity.

In the limit, completely partitioning a MECS topology yields a point-to-point network, such as the $\langle 2,4,4,3,1,1\rangle$ flattened butterfly in Figure 4.4(d). While further analysis of networks in the space of Generalized Express Cubes is beyond the scope of this paper, the experimental results in Section 4.5 include three of the networks of Figure 4.4.

4.4 Experimental Methodology

Topologies

To compare the different topologies, we used a cycle-precise network simulator that models all router pipeline delays and wire latencies. We evaluated the mesh, concentrated mesh, flattened butterfly, and MECS topologies. We also considered two topologies with replicated networks, Cmesh-X2 and MECS-X2, and a partitioned MECS variant called MECS-P2. For the largest simulated network size, we considered a variant of the flattened butterfly that limits the maximum channel span to four nodes. This topology, called FBfly4, reduces the number of bisection channels in exchange for increased per-channel bandwidth. As a result, FBfly4 enjoys a lower serialization delay compared to the regular flattened butterfly but exposes a larger network diameter.

Mesh-based topologies feature eight virtual channels per network input port; low-diameter networks have only two VCs per port with depth determined by the maximum (edge-to-edge) round-trip credit latency in an unloaded network. The relative dearth of VCs in low-diameter topologies increases the likelihood of head-of-line blocking; however, it is partly compensated for by increased path diversity in these networks. In addition, low VC counts reduce arbitration complexity, which is a prominent concern in high-radix routers.

In all configurations, we model a 3-stage router pipeline consisting of virtual channel allocation, switch allocation, and switch traversal. All topologies employ look-ahead routing, which removes route computation from the critical path. Mesh routers also benefit from speculative switch allocation, which overlaps virtual channel and crossbar allocation [56]. When successful, speculation reduces router latency to two cycles. All other topologies have a 3-cycle zero-load pipeline. We used delay models developed by Li-Shiuan Peh to estimate the latency of VC and switch allocation in the simulated high-radix routers [55]. The most aggressive design we evaluate is an 18-port flattened butterfly router in a 256-terminal network. Our analysis shows that even in this configuration, both VC and XB allocation fit comfortably in a single 20 FO4 cycle.

Network parameters

We considered network sizes of 64 and 256 terminals. Except for the mesh, all topologies use 4-way concentration, reducing the effective node count to 16 and 64, respectively. Where applicable, parameters for various topologies are the same as those in the analytical comparison of Section 4.2.3. Table 4.2 summarizes the simulated configurations.

As the bisection bandwidth across all topologies is kept constant, the concentrated mesh has twice the per-channel bandwidth of the basic mesh, while the flattened butterfly, MECS, and all replicated and partitioned topologies evenly distribute this bandwidth among their links. We double the bisection bandwidth in the 256-terminal network from that in the 64-terminal configuration to match the increase in network radix. All of the networks em-

	64 terminals	**256 terminals**
Topologies & channel BW (bits)	Mesh: 288 Cmesh: 576 Cmesh-X2 288 FBfly: 144 MECS: 288 MECS-X2: 144	Mesh: 576 Cmesh: 1152 Cmesh-X2: 576 FBfly: 72 FBfly4: 115 MECS: 288 MECS-X2: 144 MECS-P2: 144
Network organization (AxBxC) A: rows B: columns C: concentration	8x8x1: Mesh 4x4x4: Cmesh* FBfly MECS*	16x16x1: Mesh 8x8x4: Cmesh* FBfly* MECS*
Router latency (cycles)	Mesh: 2 Cmesh*, FBfly, MECS*: 3	Mesh: 2 Cmesh*, FBfly, MECS*: 3
VCs/channel	Mesh, Cmesh*: 8 FBfly, MECS*: 2	Mesh, Cmesh*: 8 FBfly* MECS*: 2
Buffers/VC	Mesh, Cmesh*: 5 FBfly, MECS*: 10	Mesh, Cmesh*: 5 FBfly*, MECS*: 18
Traffic patterns	bit complement, uniform random, transpose	
Traffic type	64- and 576-bit packets, stochastic generation	

Table 4.2: Simulated network configurations.

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

Table 4.3: Full-system configuration.

ploy dimension-order routing (DOR), resulting in paths with minimal length and hop-count.

Synthetic workloads

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

Application evaluation

To simulate full applications, we use traces from the PARSEC parallel application benchmark suite [8]. The traces were collected using the M5 full-system simulator, which was configured to model a 64-core CMP with the Alpha ISA and a modified Linux OS [9]. Table 4.3 summarizes our system configuration, comprised of two-way out-of-order cores with private L1 instruction and data caches, a shared NUCA L2 cache and eight on-die memory controllers. All benchmarks in Table 4.3 were run with sim-medium input sets with the exception of *blackscholes*, which was simulated with sim-large. The remaining PARSEC benchmarks are currently incompatible with our simulator.

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

Area and Energy

We use a combination of CACTI 6.5 [51] and custom interconnect models to estimate area and energy overheads of channels and routers for 64- and 256-terminal networks in 32 nm technology. On-chip Vdd is 0.9 V and frequency is 2 Ghz. Our interconnect model assumes intermediate-layer wires with 100 nm width/spacing (200 nm pitch), an aspect ratio of 1.8, resistance of 1.67 kΩ/mm, and capacitance of 210 fF/mm. We apply power-delay optimizations for repeater insertion described in Weste and Harris [81] to arrive at wire energy consumption of 0.052 pJ/bit/mm assuming random data. Repeaters are placed at 270 μm intervals and contribute 20% to link energy consumption. We apply this interconnect model to generate area and power profiles for links and crossbar switch fabrics. Crossbars are segmented [79] with two stages each per input and output plane. Channel wires are assumed to be routed over logic and do not contribute to link area.

For assessing buffer costs, we modify CACTI to support modestly-sized SRAM configurations with uni-directional data flow (input port to switch fabric) characteristic of router flit buffers. In MECS topologies, we model the stacked port configuration depicted in Figure 4.3. This configuration incurs an additional wire energy expense on switch traversals proportional to the height of the stack.

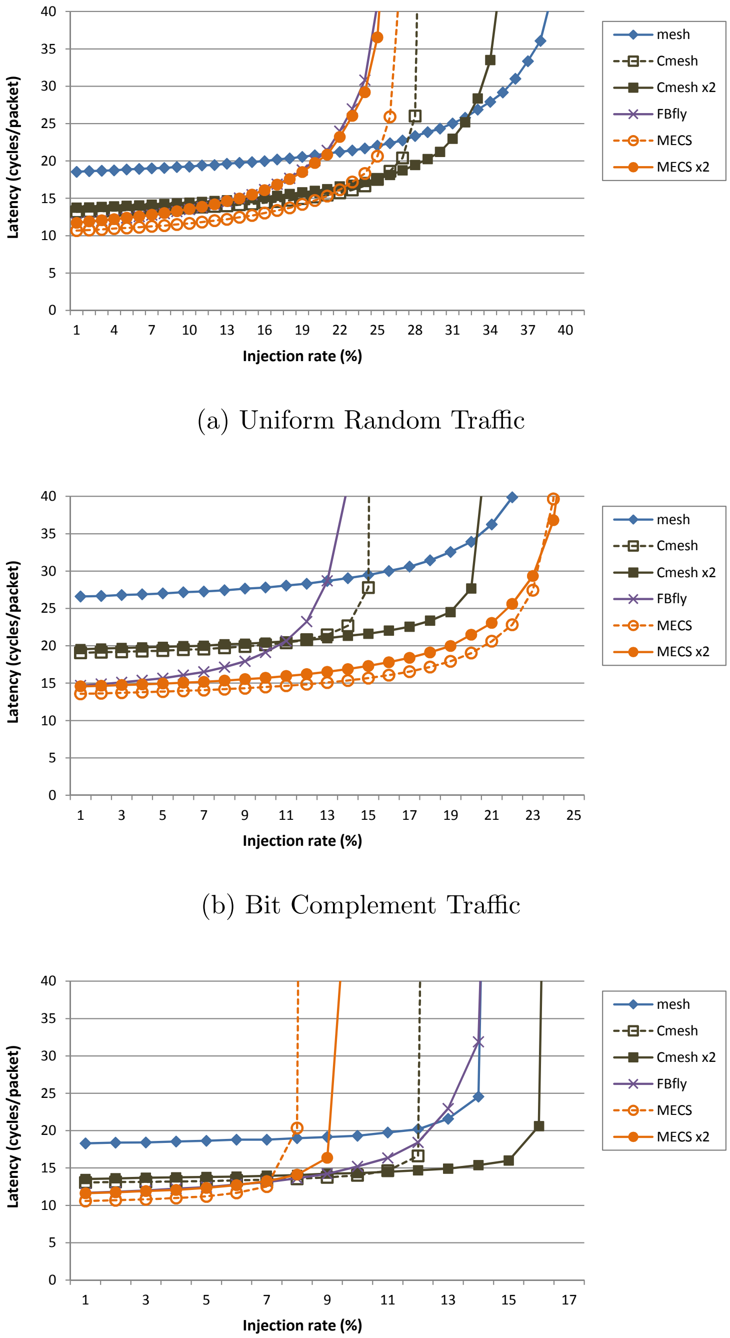
4.5 Evaluation

4.5.1 Synthetic Workload: 64 terminals

Figure 4.5 summarizes the evaluation of a 64-terminal system with the three synthetic traffic patterns. In general, we observe that the mesh has the highest latency at low loads, exceeding that of other topologies by 35-96%. The concentrated mesh variants have the second-highest latency, trailing the flattened butterfly by 12-33%. The baseline MECS topology consistently has the lowest latency at low injection rates, outperforming FBfly by 9%, on average. MECS x2 has zero-load latencies comparable to those of the flattened butterfly.

The results are consistent with our expectations. The mesh has a high hop count, paying a heavy price in end-to-end router delay. The Cmesh improves on that by halving the network diameter, easily amortizing the increased router latency. The flattened butterfly and MECS x2 have the same degree of connectivity, same number of bisection channels, and same bandwidth per channel; as such, the two topologies have similar nominal latencies. Finally, the single-channel MECS has the same connectivity as the flattened butterfly but with twice as much per-channel bandwidth, which results in the lowest zero-load latency.

The picture shifts when one considers the throughput of different topologies. The mesh, due to its high degree of pipelining, yields consistently good throughput on all three workloads. The Cmesh topology is inferior to the mesh in terms of throughput as it has fewer channels and thus supports few concurrent transfers. Cmesh x2 restores the bisection channel count lost in the baseline Cmesh due to concentration, and as a result, shows comparable throughput to the basic mesh. The flattened butterfly has the lowest throughput on two of the three traffic patterns as it cannot effectively utilize all of the available channels with dimension-order routing. MECS and MECS x2 show a high degree of variability across the different traffic patterns. We examine each traffic pattern individually to gain insight into the performance of MECS networks.



(a) Uniform Random Traffic

(b) Bit Complement Traffic

(c) Transpose Traffic

Figure 4.5: Performance of different topologies in a 64-terminal NOC.

On random traffic, MECS performance is inferior to that in mesh-based topologies due to 1) lower bandwidth out of each concentrated MECS node, and 2) the large (3:1) ratio between input and output bandwidth into each MECS router. These two factors compromise throughput by creating a load imbalance at turn nodes, where the traffic switches from rows to columns. Because random traffic is naturally balanced, every node acts as a turn node and impact on performance is limited.

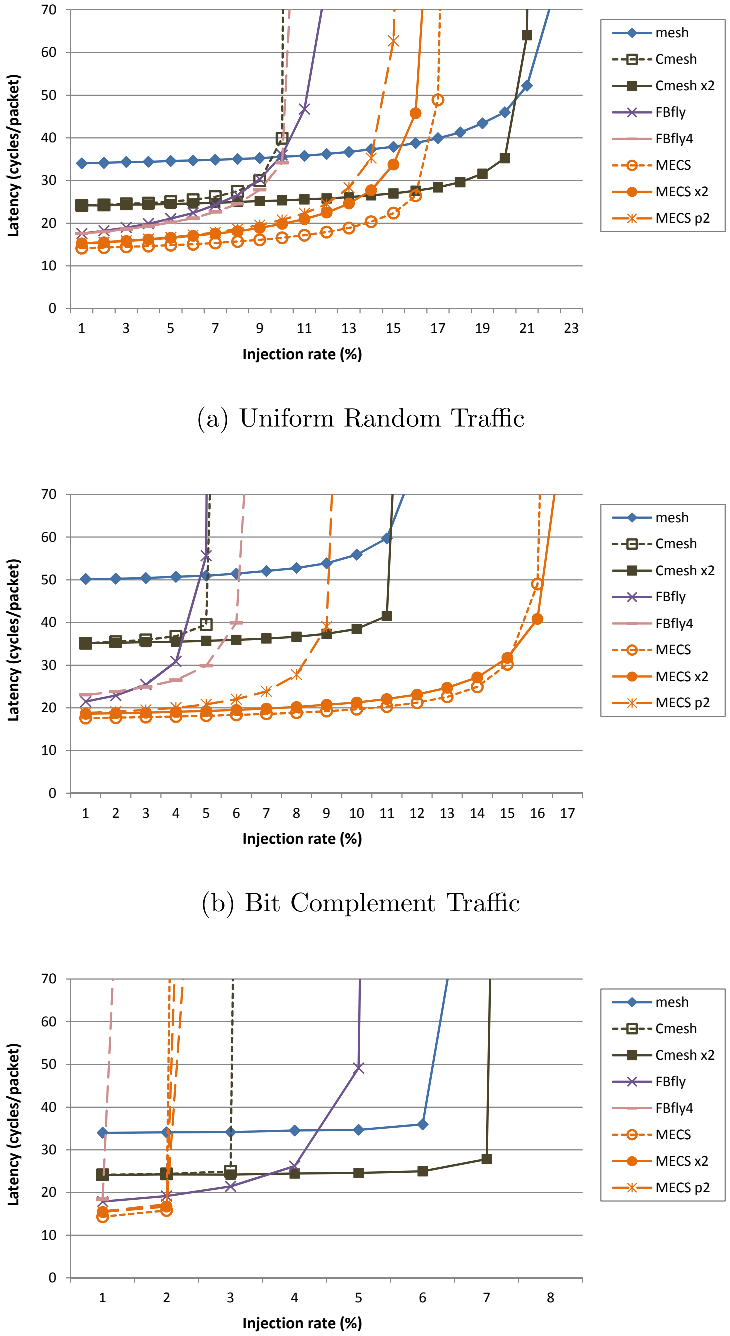
The transpose pattern, on the other hand, represents an adversarial scenario for many topologies, including MECS, under dimension-order routing. This permutation mimics a matrix transpose operation, in which all nodes from a given row send messages to the same column. In MECS, packets from different source nodes in each row arrive at the "corner" router via separate channels but then serialize on the shared outbound link, compromising throughput. The turn node is a bottleneck in mesh and Cmesh topologies as well; however, the mesh benefits from the lack of concentration, so the traffic is spread among more channels, while the Cmesh enjoys very wide channels that help throughput. Finally, the flattened butterfly achieves better throughput than MECS by virtue of its high-radix switch, which effectively provides a dedicated port for each source-destination pair. In all cases, throughput can be improved through the use of improved routing policies, which we discuss in Section 4.6.3.

On the bit-complement permutation, MECS achieves the highest performance among the examined topologies. On this traffic patter, each MECS source node communicates with its target with no contention at the intermediate router. Thus, throughput is limited only by channel width and the contention between the different terminals at each concentrated node. Traffic in the flattened butterfly topology enjoys a similar contention-free network passage; however, throughput is roughly half of that in the MECS topology since the channels are 50% narrower.

4.5.2 Synthetic Workload: 256 terminals

In the larger network, the basic mesh becomes decidedly unappealing at all but the highest injection rates due to enormous network traversal latencies, which, at low loads, are 1.4-2.8x higher than in other topologies. Cmesh also sees its latency rise significantly, exceeding that of the flattened butterfly and MECS by 35-100% at low injection rates. In all mesh-based networks, the degradation is due to the large increase in the average hop count. As expected, all MECS variants enjoy the lowest latency at low loads due to a good balance of connectivity, channel count, and channel bandwidth. As such, they outperform the flattened butterfly by 13-24% in terms of latency.

In terms of throughput, Cmesh x2 and the mesh show the highest degree of scalability. The relative performance of MECS and MECS x2 topologies follows trends similar to that in a 64-terminal network. These topologies achieve the highest throughput among evaluated networks on the bit-complement permutation; are average on random traffic; and are second-to-worst on transpose. Analysis of these performance trends in Section 4.5.1 applies here as well. The partitioned MECS, MECS p2, performs worse than MECS x2 since it has the same per-channel bandwidth as the latter but with a more restricted connectivity model. MECS p2 thus appears attractive for large networks that are sensitive to area, energy and latency but have modest bandwidth requirements. All MECS variants outperform the flattened butterfly and FBfly4 on two of the three traffic patterns. On transpose, the flattened butterfly achieves higher throughput as it is able to load all of the outgoing channels at the turn nodes. On other traffic patterns, both flattened butterfly variants saturate early, as they are unable to keep all of the channels utilized. FBfly topologies also suffer from high serialization delays induced by relatively low channel width. These delays increase the topologies' zero-load latency by 14-31% versus MECS variants.



(a) Uniform Random Traffic

(b) Bit Complement Traffic

(c) Transpose Traffic

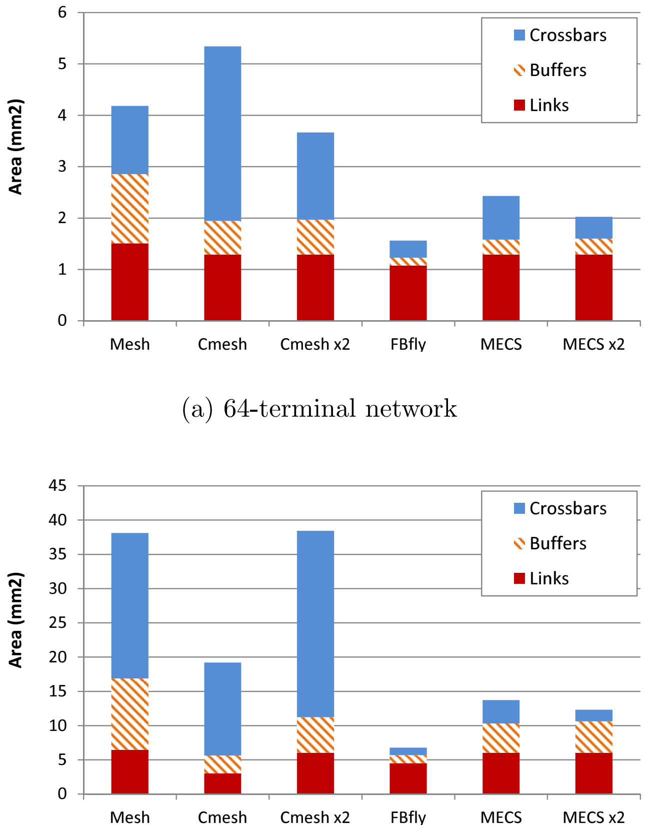
Figure 4.6: Performance of different topologies in a 256-terminal NOC.

4.5.3 Area

We assess the area-efficiency of various NOC organizations by comparing the contribution of links and routers to network area based on the methodology detailed in Section 4.4. To summarize, we assume that channel wires are routed over active areas and only include the area of drivers and repeaters when computing the link area. Our router model accounts for flit buffers and the crossbar switch fabric, which are the primary contributors to router area. Figure 4.7 shows the results of the evaluation.

In the 64-terminal network, low-diameter topologies have a 34-71% advantage in area-efficiency over mesh-based networks. The single-network Cmesh topology has the worst energy efficiency due to the large crossbar at each concentrated router. In contrast, a two-way replicated Cmesh (Cmesh x2) is the most efficient organization among meshes. Cmesh x2 reduces the switch area over the single-network Cmesh while benefiting from fewer nodes compared to the basic mesh topology. Reduction in switch area stems from the lower channel width. As noted in Section 4.2.3, crossbar area is quadratically related to port width. Since the datapath of the Cmesh x2 network is one-half as wide as that in the unreplicated Cmesh, each switch is a factor of four smaller. With two networks, switch area savings approach 2x over a single wide Cmesh.

MECS routers have the same switch complexity, as measured by the number of ports and the width of each port, as routers in Cmesh x2. However, MECS reduces buffer area over Cmesh x2 by virtue of having fewer VCs per router and avoids the area expense of a second network. As a result, MECS has a 34% lower area footprint than Cmesh x2. In turn, MECS x2 reduces switch area over the single-network MECS organization through replication, and yields the second most area-efficient organization. The lowest network area at fixed bisection bandwidth is found in the flattened butterfly. While FBfly requires more router switch ports than MECS variants (10 ports in FBfly, eight in MECS), it does not have the overhead of a second network and is 23% more area efficient than MECS x2 at the same channel width.



(a) 64-terminal network

(b) 256-terminal network

Figure 4.7: Area requirements of different NOC organizations under fixed bisection wire budget.

As we scale up the networks to support 256 terminals, we grow the bisection bandwidth to match the 2x increase in network radix. Because mesh-based topologies have relatively few channels, the available bandwidth is sufficient to provision them with 576-bit links, which is the maximum packet width in the simulated system. Since channels wider than 576 bits provide no additional benefit, the single-network Cmesh has one-half of the bisection bandwidth of Cmesh x2 and other topologies. Link width stays constant in MECS compared to the 64-terminal organization, as the increase in channel count matches the growth in bisection bandwidth. In contrast, the flattened butterfly experiences a 2x reduction in its per-channel bandwidth due to the quadrupling in channel count.

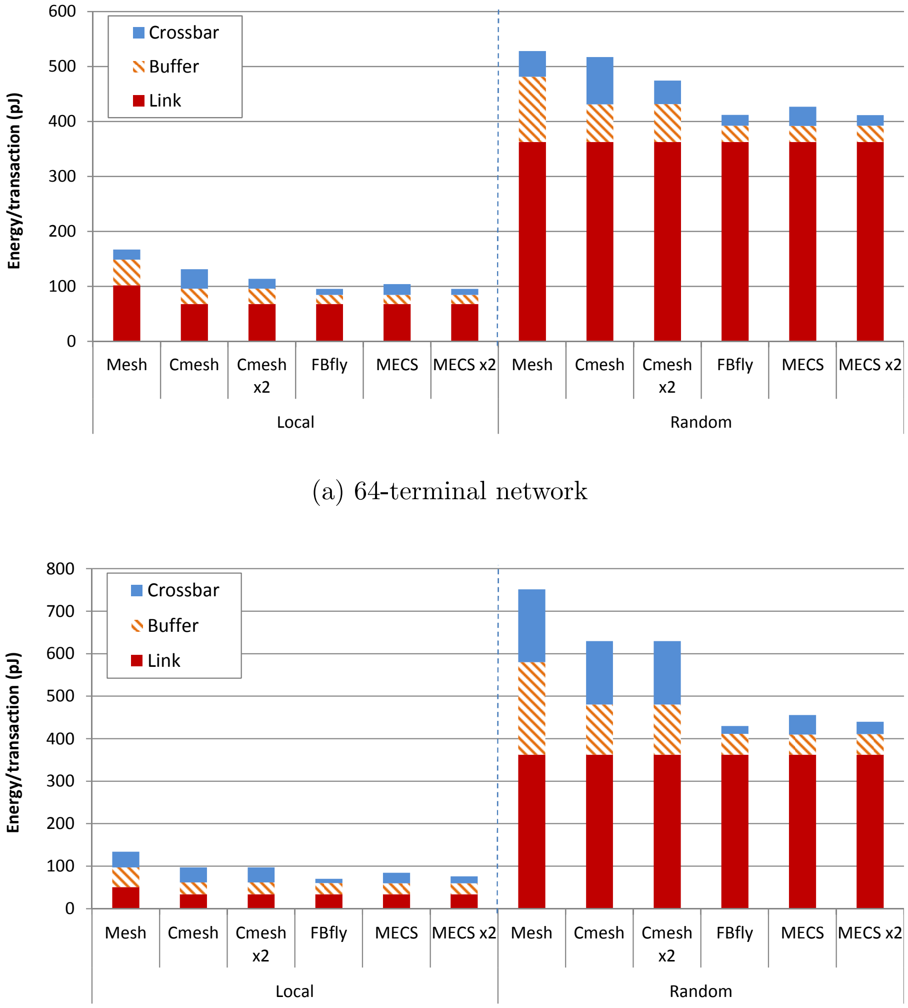
As before, mesh-based topologies are inferior to low-diameter ones in terms of area efficiency. Mesh and Cmesh x2 have a similar area footprint, which is 64-82% greater than that of low-diameter networks. The switch fabric dominates the cost of mesh-based topologies. The single-network Cmesh has one-half the bandwidth, and therefore one-half the area, of Cmesh x2. MECS is 64% more area-efficient than Cmesh x2. MECS x2 provides an additional 10% area reduction over MECS thanks to a more compact switch at each router. Finally, the flattened butterfly enjoys the lowest network area due to the smallest datapath width among the evaluated networks. The flattened butterfly reduces buffer area relative to both replicated- and unreplicated MECS topologies, as the latter have the same number of input ports at each router as FBfly, but have higher per-port bandwidth. The flattened butterfly also has lower link area than MECS variants, explained by the fact that while the center of the FBfly network has a high concentration of links, the number of links tapers off toward the edges of the network. MECS does not have such tapering and features a uniform channel count across all network cuts. Fewer channels in MECS networks translate into higher per-channel bandwidth and superior performance relative to the flattened butterfly; however, these features come at an area penalty of 51% and 45%, respectively, for MECS and MECS x2 organizations. FBfly does require a high-radix 18-port switch fabric at each router, but the area cost is offset through low per-port bandwidth.

4.5.4 Energy

To evaluate the relative energy efficiency of different topologies, we introduce the notion of a *transaction*. A transaction is composed of a 72-bit Request packet and a 576-bit Reply. We consider two types of transactions – those with high locality and no locality. Transactions with high locality communicate over a distance of one or two hops in a mesh network (zero or one hops in a Cmesh). Those with no locality have an average communication distance equal to that under random traffic, which is 5.3 (2.7) hops in a 64-terminal mesh (Cmesh) network, and 10.7 (5.3) hops in 256-terminal mesh (Cmesh) organization.

Figure 4.8(a) shows the energy per transaction for the evaluated topologies in the 64-terminal networks. The mesh is the least efficient topology, as it exposes the network diameter and requires the largest number of router traversals. Concentration reduces the network diameter and eliminates some fraction of link traversals for local traffic whose destination is the same as the source node. As a result, the Cmesh topology is 22% (2%) more efficient on local (non-local) traffic. The Cmesh x2 network has a more compact switch at each router than the unreplicated Cmesh, which helps reduce the energy expense of switch traversal. This feature improves the energy-efficiency of Cmesh x2 by 13% (8%) on local (non-local) traffic over the single-network Cmesh. MECS reduces the number of router traversals on non-local traffic over mesh-based topologies, improving energy efficiency by 10% over Cmesh x2. The savings are limited as the inter-tile wire energy dominates the router energy. When considering only the router component of energy consumption, MECS improvement in energy efficiency over Cmesh x2 is 43%. FBfly and MECS x2 are the most efficient organizations thanks to rich connectivity and modest switch size, with up to 8% (4%) improvement over MECS on local (non-local) traffic.

Similar trends are observed in the larger, 256-terminal, network. Re-



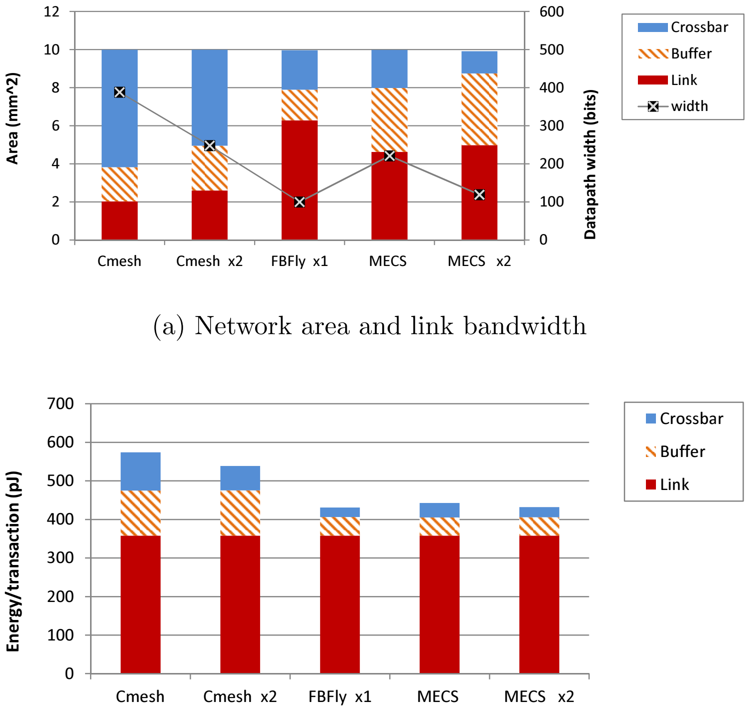
(a) 64-terminal network

(b) 256-terminal network

Figure 4.8: Energy per transaction in different NOC organizations.

sults are shown in Figure 4.8(b). The gap between mesh and express channel topologies grows as a larger fraction of the energy in mesh and Cmesh networks is expanded in routers due to increased network diameter. On non-local traffic, routers constitute 42-52% of NOC energy expense in mesh-based topologies, while contributing just 16-20% to network energy in low-diameter organizations. Cmesh and Cmesh x2, which have the same energy profile due matching datapath widths, are 28% more efficient than the unconcentrated mesh on local accesses, and 16% more efficient on non-local ones. In turn, MECS is 13% more energy-frugal on local (28% on non-local) traffic than Cmesh organizations due to richer connectivity and a narrower datapath, which reduces switch energy expense. Finally, MECS x2 and FBfly show the highest degree of energy efficiency with 17% (6%) savings in network energy over MECS on local (non-local) transactions.

The MECS crossbar energy data in Figure 4.8 includes both the switch traversal energy as well as the energy expanded in the switch input bus. This bus spans the height of the input port stack and allows the multiple network ports from a common direction to share a switch port (see Figure 4.3). The energy expanded in the bus is a function of the stack height, which, in turn, depends on the number of network ports in the stack and the depth of the flit FIFOs at each port. In a 64-terminal concentrated network, the network radix is four; as a result, up to three ports compose a stack. In the 256-terminal NOC, the radix is eight and the stack grows to seven ports. Furthermore, the larger network requires more flit buffers per port to accommodate the longer credit return delays, an attribute that necessitates deeper flit FIFOs. The end result is that the switch input bus accounts for 10% (18%) of the switch energy in the smaller MECS (MECS x2) network, while consuming 29% (45%) in the larger network with 256 terminals. While the absolute contribution of the input bus to switch energy is the same in both MECS and MECS x2 networks, the relative contribution is greater in MECS x2 due to a more compact crossbar layout, which reduces the energy expanded in the crossbar proper.



(a) Network area and link bandwidth

(b) Network energy-efficiency

Figure 4.9: Topology comparison under a fixed 10 mm^2 area budget.

4.5.5 Area-normalized Topology Comparison

Sections 4.5.3 and 4.5.4 assessed the area- and energy-efficiency of different topologies under a fixed bisection bandwidth constraint. Here, we analyze the topologies assuming a 10 mm^2 NOC area budget in a 256-terminal network. Figure 4.9(a) breaks down the NOC area by link, buffer, and crossbar costs in each of the concentrated topologies. The line, which corresponds to the secondary Y-axis, shows the datapath width for the respective configurations. Wider datapaths would increase the network area beyond the 10 mm^2 target, while narrower designs would fall short of it.

As expected, meshes feature the widest datapaths due to the limited number of channels in these topologies. However, their area is crossbar-dominated and less than 24% of the network is devoted to channels. As a result, these topologies have the lowest bisection bandwidth (not shown in the diagram). MECS has the most balanced allocation of resources across the network and affords the widest datapath (221 bits) among the low-diameter topologies. Compared to Cmesh x2 (248 bits wide), MECS provides 78% more bisection bandwidth while sacrificing just 11% in channel width. Compared to the flattened butterfly (100 bits wide), MECS has 45% less bisection bandwidth but 121% more per-channel bandwidth.

Figure 4.9(b) compares the topologies on energy-efficiency assuming uniformly distributed traffic. MECS is 18-23% more frugal than Cmesh and Cmesh x2, which is a similar advantage to that observed with topologies normalized to fixed bisection bandwidth. In contrast, the energy advantage of the flattened butterfly over MECS diminishes to just 3%, as the wider datapath reduces the efficiency of the FBfly topology.

4.5.6 Application-based Workloads

Figure 4.10 shows the relative performance and energy of various topologies on our PARSEC trace-driven workloads in a 64-terminal network. The topologies have equal bisection bandwidth. Results reflect total network energy and average per-packet latency. Because the injection rates in the simulated applications are low, latency appears to be a more stringent constraint than bandwidth for evaluating the topologies. MECS has the lowest latency, consistently outperforming the flattened butterfly and MECS x2 by nearly 10%. The mesh has by far the highest latency as a result of its high hop count, followed by the Cmesh x2 and the basic Cmesh.

Energy trends also track closely the results of our analytical model. The mesh is the least efficient organization, followed Cmesh and Cmesh x2. MECS reduces the NOC energy consumption by 10-23%, while MECS x2 provides an additional 3% energy reduction over MECS. The flattened butterfly has the same performance and energy profile as MECS x2 on these workloads.

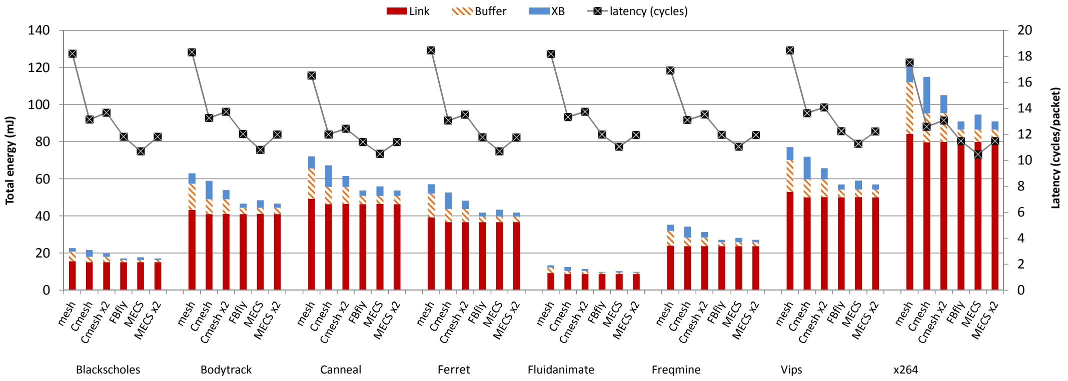


Figure 4.10: Performance and energy efficiency of different topologies on the PARSEC suite.

4.5.7 Summary

Our evaluation shows that the MECS topology provides a good balance of performance, area-, and energy-efficiency in both 64- and 256-terminal configurations. On random traffic at low network utilization, MECS reduces packet latency by no less than 19% and up to 58% over mesh-based organizations. The latency advantage is 9-19% over the low-diameter flattened butterfly topology. While maximum throughout of MECS is often lower than that of mesh networks, MECS typically bests the flattened butterfly at maximum sustained load.

The performance benefits of MECS come at modest area and energy cost. By virtue of its low network diameter, MECS incurs only a fraction of router traversals compared to mesh-based networks. As routers are responsible for considerable energy expense in NOCs, the rich connectivity reduces energy overheads in a MECS topology. MECS further limits the router energy consumption through a compact switch organization that has multiple network ports sharing a switch port. The low-radix switch design is advantageous from an area perspective as well. In general, a MECS network is 9% to 39% more energy-efficient and 26-57% more area-frugal than mesh-based organizations with the same bisection bandwidth. While a flattened butterfly has a lower area profile and somewhat better energy efficiency, the benefits come at a price of significantly lower per-channel bandwidth due to the large number of links required by the topology. MECS area and energy overheads can be reduced through replication. A two-way replicated MECS topology is able to match the flattened butterfly on energy efficiency and narrows the gap in area efficiency.

Under a fixed area budget, MECS maintains its benefits over mesh-based organizations in terms of energy efficiency. Compared to the flattened butterfly, MECS enjoys a significant advantage in per-channel bandwidth with virtually identical energy characteristics.

4.6 Discussion

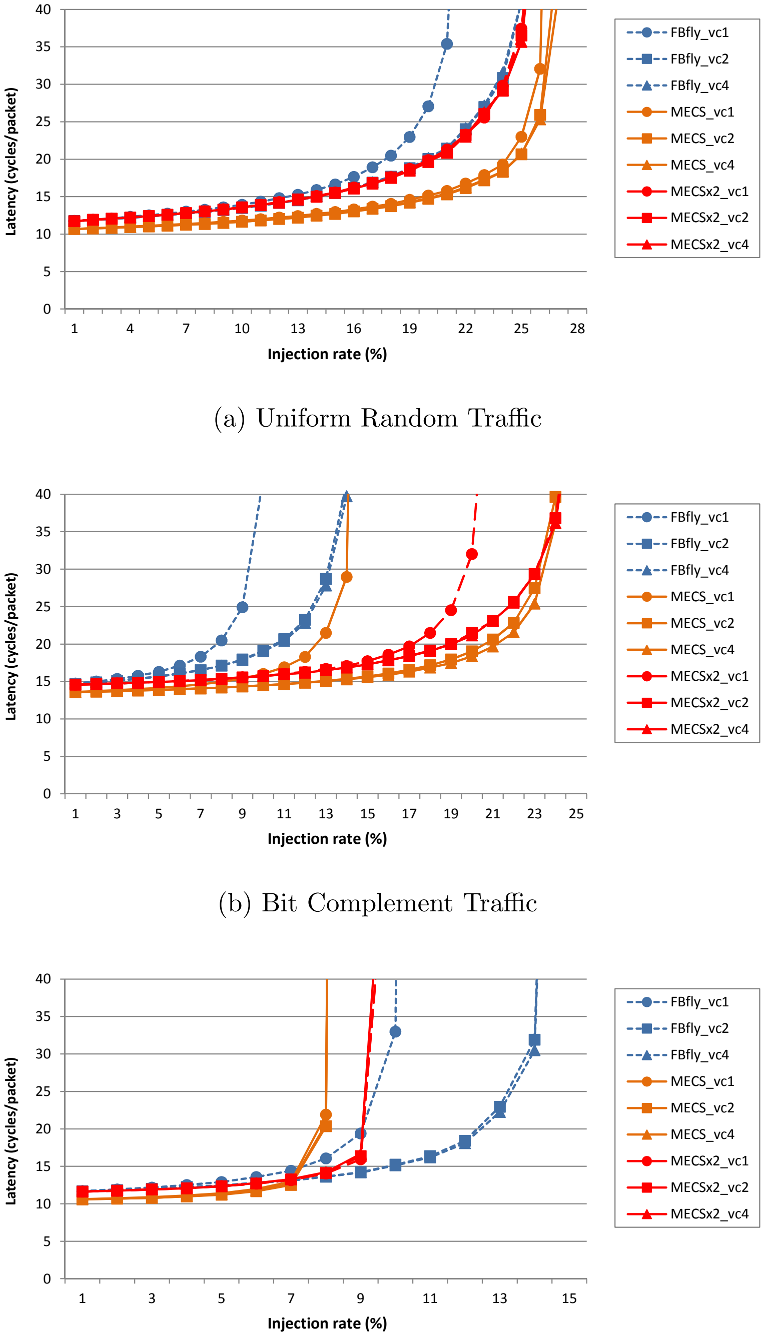
In this section, we analyze the sensitivity of the MECS topology to virtual channel organization, switch connectivity, and area budget. For brevity, we only present the results for a 64-terminal NOC; however, the trends and conclusions hold in the larger network as well.

4.6.1 VC Sensitivity

Network VCs

We first evaluate the sensitivity of MECS networks to the number of virtual channels at network interfaces. We experiment with 1, 2, and 4 VCs per network port and compare MECS, MECS x2, and FBfly topologies. The traffic patterns are *random*, *bit complement*, and *transpose*. The low-diameter networks in Section 4.5.1 used an organization with 2 VCs per port. Our analysis considers only the performance implications of virtual channels and ignores issues such as deadlock avoidance that may necessitate additional VCs.

The results are summarized in Figure 4.11. Both MECS variants appear less sensitive to the number of virtual channels than the flattened butterfly. This is explained by the fact that the switch fabric in MECS is oversubscribed



(a) Uniform Random Traffic

(b) Bit Complement Traffic

(c) Transpose Traffic

Figure 4.11: VC sensitivity of low-diameter topologies in a 64-terminal NOC.

in the sense that there are more inputs into the switch than outputs. As a result, if a packet at one input port is unable to acquire a downstream virtual channel, a packet at another port targeting a different destination may succeed in allocating a VC. As long as packets from different input ports content for the same output port but target different destination routers, the switch stays fully utilized and little benefit can be derived from multiple virtual channels. This behavior is observed in random and transpose permutations. In the bit complement pattern, on the other hand, packets from different nodes do not interfere at link level, but packets from different terminals of a given node do. As a consequence of this behavior, multiple virtual channels are advantageous under this workload. Furthermore, the benefit of multiple VCs is greater in a single-network (67% improvement in throughput over 1 VC) than in a replicated (19% improvement over 1 VC) MECS organization. Since replication enables concurrent transfer of multiple packets on disjoint networks, it partly makes up for the lack of virtual channels.

In contrast, the symmetric high-radix switch in the flattened butterfly enables concurrent transfers between all unique input-output pairs. Thus, across all traffic patterns, performance is inhibited in organizations with 1 VC per input port, as packets from different ports targeting a common destination experience frequent VC stalls. The low channel bandwidth of the flattened butterfly further exacerbates the need for multiple virtual channels due to the long serialization delays that increase the likelihood of contention.

Finally, we note that across all topologies and traffic patterns, two VCs per port is sufficient for maximum performance and virtually no additional benefit is derived in the 4 VC configuration.

Injection VCs

We also consider the sensitivity of the low-diameter networks to the number of virtual channels at the local interfaces. Results under uniform random traffic with 1 and 2 injection VCs are shown in Figure 4.12. In the legend, the number following the topology name is the number of injection VCs. The

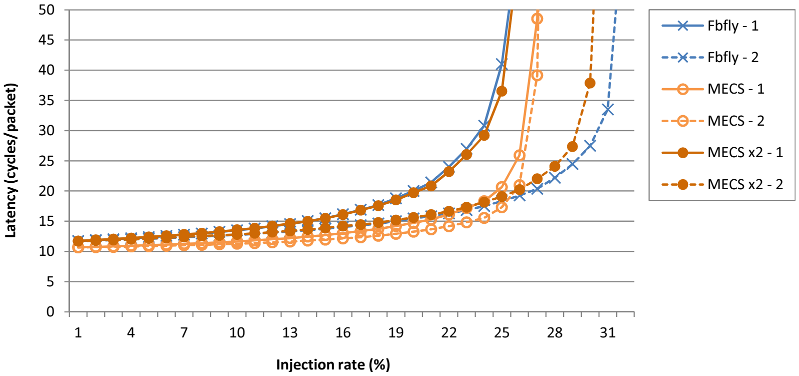


Figure 4.12: Sensitivity of low-diameter topologies to injection VC count.

baseline low-diameter networks evaluated in Section 4.5.1 featured 1 injection VC per terminal. All organizations use 2 network VCs per port.

Compared to an organization with 1 injection virtual channel, a 2-VC flattened butterfly registers a 23% improvement in throughput, MECS x2 sees a 19% benefit, while single-network MECS shows virtually no throughput gain. In the case of MECS, the benefit of a second VC is negligible due to contention for switch and output bandwidth. In MECS x2, the second network allows for concurrent packet transfers, but a single injection VC causes head-of-line blocking in cases where one network backs up due to congestion. With 2 injection VCs, a packet that stalls at an injection interface due to congestion on one network does not block packet injection from the affected terminal, as the second VC allows another packet to proceed via a different network. Similarly, in the flattened butterfly topology with 2 VCs, the high-radix switch and dedicated channels allow enable transfers to uncongested nodes in the presence of a stall to a congested destination. As before, FBfly is more sensitive to VC availability due to low per-channel bandwidth. This attribute increases the serialization bottleneck and leads to early onset of congestion, whose effect can be diminished via multiple virtual channels which help combat head-of-line blocking.

4.6.2 Switch Connectivity

Limited crossbar connectivity is one aspect of the MECS topology that may hinder its performance, as evidenced by the analysis in Section 4.6.1. To evaluate the effect of switch configuration on performance of a MECS network, we experimented with three design choices.

Baseline: The baseline switch design is the symmetric low-radix configuration with four network inputs shown in Figure 4.3 and used throughout the evaluation. This configuration has the highest degree of port over-subscription on input to the switch and is expected to provide the lowest performance.

Max: Our second configuration maximizes connectivity and features a dedicated switch input interface for each network port. In a 64-terminal concentrated network with four nodes per dimension, there will be six network input ports (three per dimension) at every router. The *max* crossbar configuration provisions each of these ports with a dedicated switch interface. The resulting switch is asymmetric, with six input and four output interfaces.

Balanced: The last design seeks to achieve a cost-performance compromise between the previous two configurations. Similar to the baseline, the *balanced* crossbar features four switch input interfaces. However, this configuration maps input ports to switch interfaces in a manner that minimizes the degree of over-subscription on switch inputs. We leverage the observation that routers near the edge of the die have fewer connected inputs from the direction of the nearby chip edge than from the opposite direction. By mapping network inputs so as to approximately balance the number of connected ports per switch interface, we hope to reduce switch pressure and improve performance. An additional benefit of this design is that the height of the input stack is reduced from $k - 1$ to $k/2$ ports.

Figure 4.13 shows the performance of the different schemes on uniform random traffic in an unreplicated MECS network. Both *max* and *balanced* switch configurations boost throughput by around 9% over the baseline de-

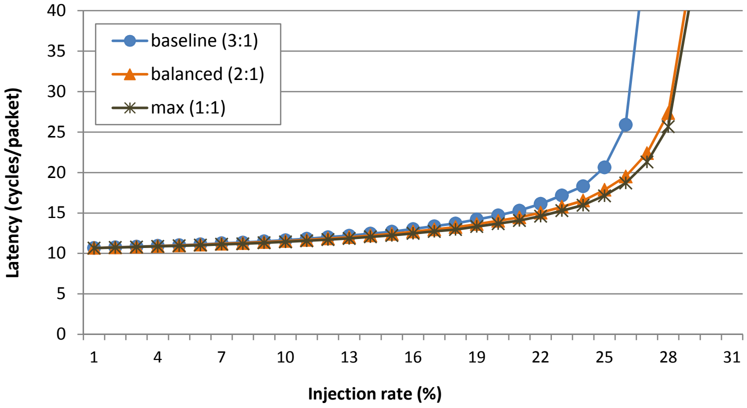


Figure 4.13: Sensitivity of MECS to crossbar connectivity.

sign. The modest degree of improvement is explained by the fact that in all configurations, the output ports are oversubscribed with respect to the input ports. Performance gains are therefore tempered by the difference in input and output bandwidth of a MECS router. In light of this finding, we conclude that a *balanced* crossbar is the most attractive design point for a MECS router, since it improves performance relative to the baseline configuration without the additional hardware overhead of the *max* scheme.

4.6.3 Adaptive Routing

Adaptive routing can improve network load balance and boost throughput by smoothing out traffic non-uniformities [68]. To evaluate its impact across the topologies, we focused on a family of adaptive routing algorithms based on O1Turn [65]. We consider both the original (statistical) approach and a number of adaptive variants that use various heuristics to estimate network congestion and choose the first dimension in which to route. Among the heuristics considered were the degree of link multiplexing, downstream VC and credit availability, and a simple variant of RCA. After evaluating each routing policy on every topology, we picked the algorithm that performed best for a given topology across all three of our synthetic traffic patterns.

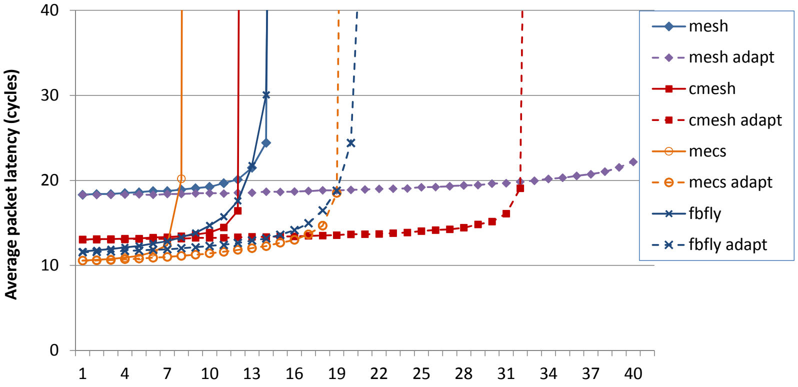


Figure 4.14: Performance of routing policies and topologies on *transpose* traffic.

The topologies see little benefit from adaptive routing on uniform random and bit-complement traffic patterns. However, Figure 4.14 shows that all topologies demonstrate a substantial throughput improvement under the transpose permutation. The mesh achieves the highest throughput relative to other topologies, as the lack of concentration allows each source to use a YX route without interference from any other node in the network. The Cmesh, despite having wider channels, does not have this luxury, as all terminals of a single concentrated node share the same set of output ports. The flattened butterfly and MECS have the same limitation as Cmesh, but also have narrower channels, thereby saturating at a lower injection rate. MECS is able to cover most of the gap relative to the flattened butterfly, almost matching its throughput.

Because the best routing algorithm is typically tied closely to the characteristics of the topology, comparing all of the topologies using O1Turn derivatives is neither complete nor completely fair. For example, the path diversity available in the flattened butterfly motivated the authors of that paper to use a non-minimal adaptive routing algorithm in their evaluation [40]. Because non-minimal routing incurs design complexity overheads and energy penalties for non-minimally routed packets, adaptive routing algorithms must balance throughput needs and energy targets of NOCs. Results on the PARSEC benchmarks suggest that these real-world applications have modest injection rates, implying that NOCs may be more sensitive to latency and energy than throughput.

4.7 Summary

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

Compared to mesh-based topologies, MECS offers a significant improvement in network latency and energy-efficiency by virtue of its rich connectivity. However, under a fixed bisection bandwidth constraint, MECS has inferior throughput under high load. Lower throughput in a MECS network is due to a mismatch in bandwidth into and out of each router, a feature that helps reduce cost and complexity. Compared to a richly-connected flattened butterfly network, MECS offers competitive energy efficiency while improving both latency and throughput. The performance benefits arise as a result of wider channels afforded in a MECS topology due to the fact that it requires fewer links versus the flattened butterfly.

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

Chapter 5

Preemptive Virtual Clock: A Flexible, Efficient, and Cost-effective QOS Scheme for Networks-on-Chip

Integration of multiple execution and storage resources on a single die has boosted processor capabilities and has even resulted in the emergence of novel usage models for computers, such as server consolidation and cloud computing. However, as the number of on-chip compute resources increases, so does the number of intra- and inter-application threads executing concurrently on a given substrate. These threads compete for shared resources, such as cache space, specialized accelerators, on-chip network bandwidth, and off-chip memory bandwidth. Ensuring application stability, scalability, and isolation in the face of significant fine-grained resource sharing is emerging as an important problem for single-chip systems.

Today's CMPs lack a way to enforce priorities and ensure performance-level isolation among the simultaneously-executing threads. For instance, in

Portions of this chapter appear in the published version of the work [32].

a cloud setting, multiple users may be virtualized onto a common physical substrate. This scenario creates a number of concerns, including inadvertent interference among the users, deliberate denial-of-service attacks, and side-channel information leakage vulnerabilities. Researchers have demonstrated a number of such attacks in a real-world setting on Amazon's EC2 cloud infrastructure, highlighting the threat posed by chip-level resource sharing [59].

At the interconnect level, a number of elegant QOS disciplines have been developed over the years to provide hard guarantees, strong isolation, and good performance. However, these mechanisms targeted conventional networks characterized by a different set of constraints as compared to single-chip systems. In an on-chip setting, these approaches incur significant area, energy, and delay overheads due to their high buffer requirements and complex scheduling policies.

In this chapter, we seek to understand the qualities of an ideal QOS solution for networks-on-a-chip (NOCs). We draw on traditional QOS literature and supplement it with our own observations to enumerate the attribute set of an ideal NOC QOS scheme. Our insights lead us to propose Preemptive Virtual Clock (PVC), a novel QOS scheme specifically designed for cost- and performance-sensitive on-chip interconnects. Unlike all prior approaches for providing network quality-of-service, PVC requires neither per-flow buffering in the routers nor large queues in the source nodes. Instead, PVC provides fairness guarantees by tracking each flow's bandwidth consumption over a time interval and prioritizing packets based on the consumed bandwidth and established rate of service. PVC avoids priority inversion by preempting lower-priority messages. The system provides guarantees and low latency for preempted messages via a dedicated ACK/NACK network and a small window of outstanding transactions at each node. Unique to this approach is the ability to trade the strength of throughput guarantees of individual flows for overall system throughput. Finally, PVC simplifies network management by enabling per-thread, per-application, or per-user bandwidth allocation.

The rest of the chapter is structured as follows. Section 5.1 motivates the work by outlining the requirements for NOC QOS techniques and presents an overview of prior approaches for network quality-of-service. Section 5.2 introduces PVC and compares it to prior schemes based on the attributes from Section 5.1. Section 5.3 covers the evaluation methodology, while Section 5.4 presents the results of the evaluation. Section 5.5 characterizes PVC under different topologies and flow control regimes. Section 5.6 concludes the chapter.

5.1 Motivation

5.1.1 NOC QOS Requirements

An ideal NOC QOS solution should possess a number of attributes with regard to guarantees, performance and cost. In this section, we draw on traditional QOS literature and supplement it with our own observations to detail the desirable feature set. Items *a, b, c, e, i, j* are taken from or inspired by a similar list compiled by Stiliadis and Varma [71], while *f* comes from Demers et al. [17].

a) *Fairness:* Link bandwidth must be divided among requesting flows equitably based on individual reserved rates for both guaranteed and excess service.

b) *Isolation of flows:* Rate-observing flows should enjoy the illusion of a private network with bandwidth proportional to the specified rate, regardless of the behavior of other flows.

c) *Efficient bandwidth utilization:* Flows should be free to claim idle network bandwidth regardless of their reserved rate or bandwidth usage history.

d) *Flexible bandwidth allocation:* It should be possible to allocate bandwidth at granularity of a core, a multi-core application, or a user. Coarser granularities simplify provisioning and improve bandwidth utilization.

e) *Low performance overhead:* Compared to a similarly provisioned network with no QOS support, a QOS-enabled network should enjoy approximately equal latency and overall throughput.

f) *Delay proportional to bandwidth usage:* Flows that observe their assigned bandwidth should enjoy faster service than flows that exceed their bandwidth ceiling.

g) *Low area overhead:* Per-flow buffering at each network node may be too expensive for on-chip networks that typically feature wormhole switching and a small number of virtual channels.

h) *Low energy overhead:* Energy may be the biggest constraint in future CMPs and SOCs [54]. Minimizing buffering is one way to reduce the energy overhead of a QOS subsystem.

i) *Good scalability:* As the network is scaled up in size, the QOS subsystem should be easy and cost-effective to scale proportionately, without compromising performance or guarantees.

j) *Simplicity of implementation:* Design and verification time are important contributors to overall system cost, and a simpler QOS solution is generally preferred to one with greater complexity.

5.1.2 QOS Service Disciplines

A number of distinct disciplines have emerged over the years for providing fair and differentiated services at the network level. We partition these into three classes based on their bandwidth allocation strategy – *fixed, rate-based, and frame-based* – and cover the most important representatives of each class.

Fixed bandwidth allocation

Approaches such as Weighted Round Robin use a static packet schedule to deliver hard guarantees at low implementation complexity. The cost, however, is potentially poor network utilization, as resources held by idle flows cannot be rapidly redistributed to flows with excess demand.

Rate-based approaches

Rate-based service disciplines aim to allocate bandwidth to contending packets based on the provisioned rate. Idle bandwidth due to under-utilization by one or more flows is instantaneously redistributed among the competing flows. Service order is determined dynamically based on the set of active flows and their respective reserved rates by computing the service time for each flow and granting the flow with the earliest deadline. In general, rate-based approaches excel at maximizing throughput and providing strong isolation, but necessitate per-flow queueing and may require computationally expensive scheduling algorithms.

Fair Queueing (FQ) is a well-known rate-based approach that emulates a bit-by-bit round-robin service order among active flows on a packet basis [17]. Its generalized variant, Weighted Fair Queueing (WFQ), enables differentiated services by supporting different service rates among the flows. Both schemes offer provably hard fairness guarantees at a fine granularity and excellent bandwidth utilization. Unfortunately, computing the service time in FQ has O(N) complexity, where N is the number of active flows at each scheduling step, making the algorithm impractical for most applications.

In contrast, Virtual Clock [82] offers a simple deadline computation that emulates a Time Domain Multiple Access (TDMA) scheduler but with ability to recycle idle slots. Packets in a Virtual Clock system are scheduled using virtual time slots that are assigned based on the provisioned service rate. Packet service time is simply its flow's virtual clock value, which is incremented every time the flow is serviced. In flows that respect the reserved rate, termed *rate-conformant flows*, virtual time tracks the service time under TDMA. Flows can exceed the specified service rate and "run ahead" of schedule by incrementing their virtual clock beyond the current round. Problematically, flows that run ahead are subject to starvation by rate-conformant flows until the rate-scaled real time catches up with their virtual clock. Additionally, routers implementing either Virtual Clock and Fair Queueing require per-flow queues and a sorting mechanism to prioritize flows at each scheduling step, resulting in high storage overhead and scheduling complexity in networks with a large number of flows.

Frame-based approaches

Whereas rate-based disciplines aim for tight guarantees at a fine granularity by scheduling individual packets, frame-based approaches seek to reduce hardware cost and scheduling complexity by coarsening the bandwidth allocation granularity. The common feature of these schemes is the partitioning of time into epochs, or frames, with each flow reserving some number of transmission slots within a frame. A disadvantage of frame-based disciplines lies in their coarse throughput and latency guarantees, which apply only at the frame granularity. Coarse-grained bandwidth allocation can cause temporary starvation of some flows and high service rate for others, making jitter guarantees impossible. Frame-based approaches also require per-flow buffering at each routing node, necessitating enough storage to buffer each flow's entire per-frame bandwidth allocation. Schemes such as Rotating Combined Queueing (RCQ) [41] that support multiple in-flight frames to improve network bandwidth utilization incur additional area and energy overheads in the form of even greater buffer requirements.

Globally Synchronized Frames (GSF) is a frame-based QOS approach recently proposed specifically for on-chip implementation [45]. GSF also employs a coarse-grained bandwidth reservation mechanism. However, it moves the buffering and much of the scheduling logic from the network routers into the source nodes, thereby reducing the routers' area and energy overhead. Source nodes in GSF tag new packets with a frame number and slot them into their source queue. GSF supports bursts by allowing packets from future frames to enter the network, up to a maximum allowed burst size. A fast barrier network synchronizes the frames over the entire chip by detecting when the

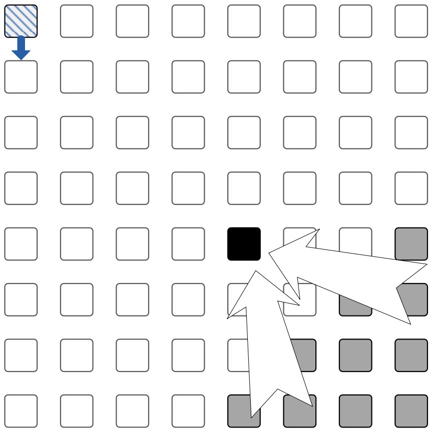


Figure 5.1: Scenario demonstrating poor bandwidth utilization with GSF.

head frame has been drained and signaling a global frame roll-over. To ensure fast frame recycling, injection of new packets into the head frame is prohibited. Packets from multiple frames may be in the network at the same time, and age-based arbitration on the frame number is used to prioritize packets from older frames over younger ones. GSF does not specify the service order within a frame, preventing priority inversion by reserving a single virtual channel (VC) at each input port for the head frame; however, in-flight packets from future frames may be blocked until their respective frames become the oldest.

Although GSF significantly reduces router complexity over prior approaches, it suffers from three important shortcomings that limit its appeal: performance, cost, and inflexible bandwidth allocation.

The performance (throughput) limitations of GSF arise due to its source-based bandwidth reservation mechanism. With only limited support for excess service, bound by the finite number of in-flight frames, GSF is inherently restricted in its ability to efficiently utilize network bandwidth. Once a source node has exhausted its burst quota, it is immediately throttled and restricted to its reserved allocation in each frame interval.

Figure 5.1 highlights a scenario that compromises a node's throughput despite idle network capacity. A set of nodes, in grey, all send traffic to a

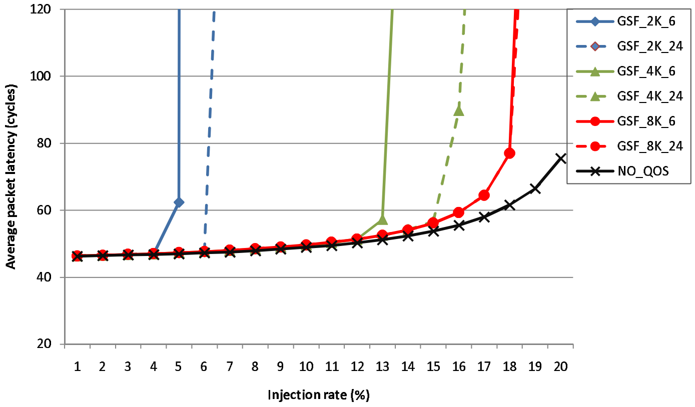


Figure 5.2: Performance of GSF with various frame (first number in legend) and window (second number) sizes versus a similarly provisioned network without QOS support.

common destination, colored black. The combined traffic causes congestion around the black node, exerting backpressure on the sources and impeding global frame reclamation. As frame reclamation slows, an unrelated node, striped in the figure, in a different region of the network suffers a drop in throughput. The striped node is only sending to its neighbor, yet is throttled upon exhausting its burst quota, letting the requested link go idle. We simulated this scenario on a 64-node network with an aggressive GSF configuration (2000 cycle frame, 6-frame burst window, and 8 cycle frame reclamation) and equal bandwidth allocation among nodes, under the assumption that the actual communication pattern is not known in advance. We observed that throughput for the striped node saturates at around 10%, meaning that the link is idle 90% of the time. Increasing both the size of the frame and the burst window ten-fold made no difference in actual throughput once the striped node exhausted its burst capacity.

Another drawback of GSF is the cost associated with the source queues, where packets are slotted to reserve bandwidth in future frames. Longer frames better amortize the latency of barrier synchronization and support bursty traffic, but necessitate larger source queues. Our experiments, consistent with results in the original paper, show that in a 64-node network, a frame size of 1000 flits or more is required to provide high throughput on many traffic patterns. To support asymmetric bandwidth allocation, whereby any node may reserve a large fraction of overall frame bandwidth, source queues must have enough space to buffer at least a full frame worth of flits. Assuming a frame size of 1000 flits and 16-byte links, GSF requires a 16 KB source queue at each network terminal. Scaling to larger network configurations requires increasing the frame size and source queues in proportion to the network size.

Figure 5.2 shows the performance of GSF under the *uniform random* traffic pattern on a 256 node network with different frame lengths and window sizes (number of in-flight frames). To reach a level of throughput within 10% of a generic NOC network with no QOS support, GSF requires a frame size of 8000 flits, necessitating 128 KB of storage per source queue.

Finally, GSF is inflexible in its bandwidth allocation, as bandwidth may only be assigned at the granularity of individual nodes, complicating network management. For instance, a parallel application with a fluctuating thread count running on multiple cores can cause a network to be reprovisioned every time a thread starts or ends, placing a burden on the OS or hypervisor.

5.2 Preemptive Virtual Clock

Our motivation in designing a new QOS system is to provide a cost-effective mechanism for fairness and service differentiation in on-chip networks. Primary objectives are to minimize area and energy overhead, enable efficient bandwidth utilization, and keep router complexity manageable to minimize delay. Another goal is to simplify network management through a flexible bandwidth reservation mechanism to enable per-core, per-application, or per-user bandwidth allocation that is independent of the actual core/thread count. This section details the resulting scheme, which we term Preemptive Virtual Clock (PVC).

5.2.1 Overview

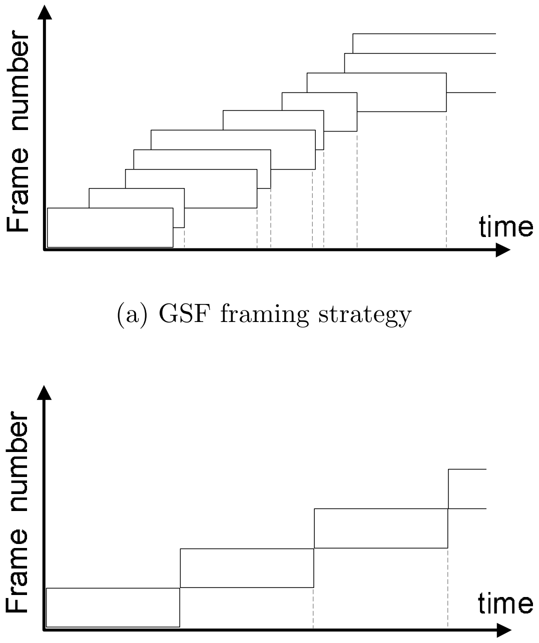
Bandwidth allocation

As the name implies, PVC was partly inspired by Virtual Clock due its rate-based nature and low scheduling complexity. Each flow in PVC is assigned a rate of service, which is translated into a certain amount of *reserved* bandwidth over an interval of time. Routers track each flow's bandwidth utilization, computing a packet's priority based on its respective flow's bandwidth consumption and assigned rate. The packet with the highest priority at each arbitration cycle receives service. Similar to Virtual Clock, flows may consume bandwidth beyond the reserved amount, potentially subjecting them to subsequent starvation from rate-conformant flows. This problem arises as a result of relying on past bandwidth usage in priority computation.

To reduce the history effect, PVC introduces a simple framing strategy. At each frame roll-over, which occurs after a fixed number of cycles, bandwidth counters for all flows are reset. Thus, PVC provides bandwidth and latency guarantees at frame granularity but uses rate-based arbitration within a frame. Because flows are free to consume idle network bandwidth, PVC does not require multiple in-flight frames or sophisticated frame completion detection to achieve good throughput. Figures 5.3(a) and 5.3(b) compare the framing schemes of GSF and PVC, respectively. GSF supports multiple in-flight frames whose completion time is determined dynamically via a global barrier network that detects when all packets belonging to a frame have been delivered. In contrast, PVC has only one fixed-duration frame active at any time. Packets in PVC are not bound to frames, and a given packet may enter the network in one frame interval and exit in the next.

Freedom from Priority Inversion

PVC uses relatively simple routers with a small number of virtual channels per input port. Without per-flow queueing, packets from flows that exceed their bandwidth allocation in a frame may block packets from rate-conformant flows.



(a) GSF framing strategy

(b) PVC framing strategy

Figure 5.3: Comparison of GSF and PVC framing strategies. GSF features multiple in-flight frames whose duration is determined dynamically via a global barrier network. PVC has a single in-flight frame of fixed duration.

Similarly, flows that greatly exceed their assigned rate may impede progress for flows that surpass their allocation by a small margin. Both situations constitute *priority inversion*. PVC uses a preemptive strategy to deal with such scenarios, removing lower priority packets from the network, thus allowing blocked packets of higher priority to make progress.

To support retransmission of dropped packets, PVC requires a preemption recovery strategy. One option for preemption recovery is a timeout. Establishing a safe timeout interval is often difficult, however. Additionally, timeouts necessitate large source buffers to support a sufficient number of outstanding transactions to cover the timeout delay. Instead, we choose to use a dedicated non-discarding ACK network for signaling packet completion and preemption events. The cost of such a network is low as its width is small compared to the wide on-chip data network. In addition, this cost may be amortized by integrating the network with the chip's fault-tolerance logic to provide end-to-end data delivery guarantees, which may be required as substrates get less reliable due to technology scaling.

As packets are subject to discard, they must be buffered at the source until an acknowledgement from the destination is received. In the case of dropped packets, preemption of the header flit generates a NACK message to the source node. Once received at the source, the NACK triggers a retransmission of the dropped packets. Thus, PVC requires a small source window to buffer outstanding transactions. Advantageously, a *small* window size acts as a natural throttle, or rate-controller, preventing individual nodes from overflowing the network's limited buffering. The window only needs to be big enough to support high throughput when the interconnect is congestion-free and allows for prompt ACK return. In our experiments, a 64-node network sees little benefit from source windows larger than 30 flits on most traffic patterns. As the network size is scaled up, the window size must increase in proportion to the network diameter to cover the longer ACK round-trip time. In a mesh topology, the diameter is proportional to the square root of the mesh size; thus, quadrupling a PVC network from 64 to 256 nodes requires doubling the source window to 60 flits.

Researchers have previously studied the use of preemption to overcome priority inversion in interconnection networks. Knauber and Chen suggest its use in wormhole networks for supporting real-time traffic [42]. Their work, however, does not consider impact on fairness, overall throughput, and recovery mechanisms. Song et al. also propose using preemption for real-time traffic [69]. Their scheme requires a dedicated FIFO at each router node where preempted packets are stored. The FIFO must have enough buffering to store a full-sized packet for each supported priority level, except the highest, requiring a significant storage overhead in systems with a large number of priority levels. Their work also does not consider fairness and other QOS-related issues.

Flow Tracking and Provisioning

Finally, PVC routers must track each flow's bandwidth utilization for scheduling and preemption purposes. While this requires additional storage, the presence of per-flow state at each router offers important advantages in network provisioning and bandwidth utilization. For instance, several threads from an application running on multiple cores can share the same flow identifier. The ability to combine multiple flows into one enables per-application bandwidth allocation, reducing management overhead when the thread count changes over the lifetime of the application. In addition, coarser bandwidth allocation granularity enables better bandwidth utilization by allowing communication-intensive threads of an application to recover idle bandwidth from less active threads.

5.2.2 QOS Particulars

Preemption Throttling

A common definition of priority inversion in a network is the presence of *one or more* packets of lower priority at a downstream node, impeding a higher priority packet's progress. A PVC system based on this definition experiences very high preemption rates under congestion, considerably degrading throughput as a result. To address this problem, we use an alternative definition that potentially sacrifices some degree of fairness in exchange for improved throughput. Specifically, priority inversion in a PVC network occurs when a packet cannot advance because *all* buffers (virtual channels) at the downstream port are held by packets of lower priority. Thus, as long as one or more downstream VCs belong to a packet of same or higher priority as the current one, preemption is inhibited. In addition, PVC employs three mechanisms for further controlling preemption aggressiveness and balancing fairness with throughput.

The first mechanism is the allocation of some *reserved bandwidth* per flow per each frame interval. The amount of reserved bandwidth, in flits, is a function of the frame size and the flow's reserved rate. Any flit within the reserved envelope is not subject to preemption, forming the basis for PVC's bandwidth guarantee.

The second mechanism for preemption throttling is based on reducing the resolution of bandwidth counters by masking out some number of lower-order bits via a programmable *coarsening mask*. Doing so reduces the resolution of the computed priority values, effectively forming coarser priority classes. Packets that map to the same priority class may not preempt each other.

The final preemption control technique built into PVC addresses a pathological case in which multiple retransmissions of a packet reduce a flow's priority by incrementing the bandwidth counters up to the preemption point. With each unsuccessful transmission attempt, the flow's priority is further reduced, compromising throughput. To avoid this pathology, PVC transmits the hop count up to the preemption point as part of the NACK sent back to the source node. In turn, the source embeds the count in a dedicated field of the retransmitted packet. This counter is decremented at each hop until it reaches zero and inhibits the update of the flow's bandwidth counter as long as it is non-zero.

Guarantees

PVC is able to make four important guarantees: minimum bandwidth, fairness, worst-case latency, and in-order delivery for rate-compliant flows. The last guarantee requires a deterministic routing function. In order for these guarantees to be met, a PVC system must comply with the following requirements:

1. No link in the system is overbooked. Thus, for every link, the sum of provisioned rates across all flows does not exceed 100%.
2. The number of reserved flits for each flow is no less than the size of the source window used for buffering outstanding transactions.

3. Resource arbitration collisions (multiple requesters with the same priority) are broken fairly (e.g., randomly). Similarly, when multiple packets at an input port have the same priority and one must be preempted, the selection mechanism is fair.

The OS or hypervisor must satisfy the first two requirements whenever the network is configured and rates are assigned to flows. The last requirement is ensured at design time. Note that the first requirement does not prevent flows from exceeding the assigned rate whenever idle network bandwidth is available, as rate enforcement occurs only under contention.

Minimum bandwidth: Each PVC flow gets a certain number of reserved flits, computed as a fraction of the frame size based on the flow's negotiated rate. These flits are not preemptable. They also cannot be blocked by packets from other flows that have exhausted their bandwidth reserve in the current frame, as preemption guarantees freedom from priority inversion. Finally, per the first requirement above, no link in the system is overbooked. Thus, all reserved flits that enter the system by the start of the frame are guaranteed to be delivered by the end.

Fairness: A PVC network distributes excess bandwidth in a fair, rate-proportional manner, choosing the flow with the lowest *relative throughput* (rate-adjusted bandwidth utilization) at each arbitration event. To resolve resource conflicts, PVC uses fair (per requirement 3) priority arbiters, described in Section 5.2.3. The degree of fairness and the strength of bandwidth guarantees is a function of the resolution of the bandwidth counters used in priority computation.

Worst-case Latency: Once a packet enters a source window, PVC guarantees its delivery by the end of the following frame interval. The guarantee is a direct outcome of requirement 2 and the minimum bandwidth guarantee. In essence, any packet in the source window will be within the reserved bandwidth cap in the new frame, thus assuring its delivery in that frame.

In-order delivery Rate-compliant flows in a PVC-enabled system with a deterministic routing function can enjoy in-order delivery, a guarantee based on two factors. The first is non-preemption of rate-compliant traffic by other flows, due to the reserved bandwidth quota in each frame interval under which all rate-compliant packets fall. The second is order preservation among packets from the same flow, which arises as a result of the monotonicity of the priority function. Priorities under PVC monotonically decrease over the duration of a frame because they are inversely proportional to bandwidth utilization. Therefore, for any pair of packets within a flow at a common arbitration point, the younger packet's priority is guaranteed to be no greater than that of the older packet. The fairness requirement, which dictates that higher priority packets are granted network resources ahead of packets with lower priority, allows older packets to proceed ahead of younger ones. To preserve relative priorities across frame boundaries, packets can be tagged with an injection time stamp or sequentially numbered at the source. Arbiters can than use these additional hints to break ties among packets with equal priorities.

5.2.3 Microarchitecture

Our baseline is a generic NOC router with no QOS support, described in Section 2.2. Its three-stage pipeline consists of virtual channel allocation (VA), crossbar allocation (XA), and crossbar traversal (XT). Figure 5.4 shows the modifications to this baseline design required to support PVC. Compared to the baseline, a PVC router needs priority computation logic, which includes the per-flow bandwidth counters and reserved rate registers. It also requires priority arbiters for virtual channel and switch arbitration instead of priority-oblivious matrix arbiters in the baseline router. Finally, a PVC router needs a preemption mechanism. All of these components are detailed next.

Priority computation logic

To support per-flow bandwidth tracking and rate-based arbitration, PVC routers must maintain per-flow bandwidth counters for each output port. In addition, each flow needs a reserved bandwidth register and a rate register,

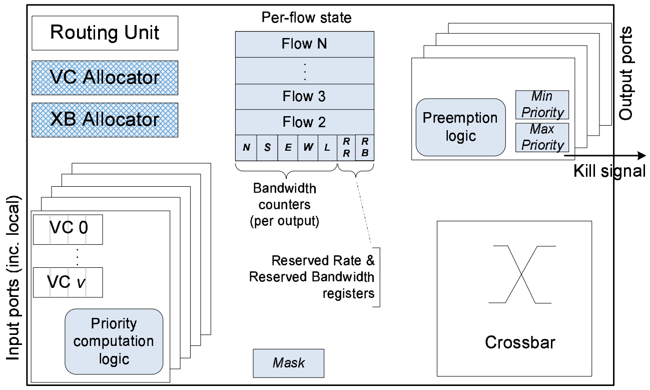


Figure 5.4: PVC router microarchitecture. Highlighted structures are new; crosshatched structures are modified relative to the baseline. *Italics* indicate a register name.

which can be shared across the ports. Finally, one *mask* register per router stores the bandwidth counter coarsening mask.

When a packet header arrives at a router's input port, priority computation logic uses the packet's flow identifier to access the bandwidth counter at the requested output port (computed at the previous hop). The access is a read-modify-write operation that increments the counter by the size of the packet, in flits. Concurrent with the update, the pre-incremented counter value is masked using the bandwidth counter coarsening mask and scaled by the flow's rate register. The resulting priority value is used for virtual channel and crossbar arbitration in subsequent cycles.

Unfortunately, the above approach adds a new pipeline stage for priority computation, increasing router delay. To remove priority calculation from the critical path, we propose using the priority value computed at the previous hop for virtual channel arbitration in the first cycle at a given node. Concurrent with VC allocation, the flow updates its bandwidth counter and priority. The updated priority is then used for any subsequent VA retries and switch arbitration requests.

The resulting approach is safe if each source node has a unique flow identifier, as the flow's bandwidth utilization at the previous node is guaranteed to be no less than its usage through any output port at the current node. In other words, the new priority can never be lower than that of the previous hop. However, this technique is not safe if multiple sources share the same flow identifier, as the guarantee breaks down under a convergent traffic pattern. Fortunately, we can still use this approach with a minor modification: if a flow wins virtual channel arbitration in its first cycle but the computed priority is lower than the value used for arbitration, the winning request is not granted and must rearbitrate with the updated priority.

Priority arbiter

Allocation delay frequently determines the router's clock frequency in conventional networks, necessitating fast arbiters. PVC benefits from not requiring per-flow buffering, which keeps arbitration complexity modest even as the network size is scaled up. At the core of our arbiter is a low-latency comparator design proposed by Harteros and Katevenis, which uses a binary comparison tree with several acceleration techniques based on fast adder circuits [34]. We anticipate that a single-cycle priority arbiter based on this comparator design can be realized for NOC networks that have up to 64 virtual channels per router.

Preemption mechanism

To support preemption, PVC requires a modification to the virtual channel allocator that enables it to assign a VC to a requester even when none of the VCs at a downstream port are free. For that purpose, PVC maintains *Min priority* and *Max priority* registers at each output port, corresponding to the downstream virtual channel with the minimum and maximum priority value, respectively. In parallel with virtual channel arbitration, each requester's priority is compared to the value of the *Max priority* register. If the requester's priority exceeds *Max priority*, the virtual channel corresponding to *Min prior-*

Feature	WFQ	GSF	PVC
a) Fairness	+	+	+
b) Isolation	+	o	o
c) Bandwidth utilization	+	o	+
d) Flexible bandwidth allocation granularity	+	-	+
e) Performance overhead	o	+	+
f) Delay proportional to bandwidth usage	+	-	+
g) Area overhead	-	-	+
h) Energy overhead	-	o	o
i) Performance scalability	+	o	o
j) Implementation complexity	o	+	o

Table 5.1: Feature comparison of QOS schemes. '+' indicates *good*, 'o' is *fair*, and '-' is *poor*.

ity is tentatively marked for preemption. VA logic assigns this virtual channel to the winning requester if none of the legal VCs are free. Of course, any packet within the reserved bandwidth envelope is not eligible for preemption.

In the next cycle, while the winning VC arbitrates for crossbar access, the resources associated with the preempted packet are released at the current node. If some part of the preempted packet has already been transferred, preemption logic sends a *kill* signal to the downstream node over a dedicated wire. The process of releasing resources held by the packet is repeated at each downstream hop until the header flit is encountered. Preemption of the header flit generates a NACK message to the source, which triggers a retransmission of the message.

5.2.4 Comparison to Prior Approaches

Table 5.1 compares three QOS schemes – WFQ, GSF, and PVC – on the feature set presented in Section 5.1.1. WFQ has excellent fairness guarantees and strong performance isolation that scale well with network size. However, it requires per-flow queueing and complex scheduling, resulting in large area and energy cost, with potentially high per-hop latency.

GSF, on the other hand, has simple routers and modest frame management hardware, yielding low router delay and low implementation complexity. However, by pushing much of the scheduling responsibility into the terminals, GSF sacrifices throughput and has no flexibility in its bandwidth allocation. GSF's other shortfall lies in its poor suitability to fine-grained communication, as our experimental evaluation in Section 5.4 confirms. Because injection into the head frame is disallowed, the scheme introduces additional latency under contention. Thus, delay is unrelated to bandwidth usage. In fact, aggressive senders can temporarily block network access to sources with low injection rates, making the scheme susceptible to a denial-of-service attack.

PVC has good bandwidth efficiency, modest router complexity and low area overhead. A shortcoming of PVC compared to GSF is PVC's higher implementation complexity, which stems from the distributed protocols associated with preemption and ACK/NACK handling, as well as the logic for per-flow bandwidth tracking at each router node.

Both PVC and GSF provide only *fair* isolation of flows, which stems from their lack of per-flow buffering at each router node. They also have some undesirable energy overheads. In PVC, the overhead results from retransmission of packets, flow table lookups, and the ACK network; in GSF, it is from source queue accesses. Finally, both approaches leave room for improvement with regard to performance scalability. As the network size is scaled up, GSF becomes increasingly prone to bandwidth coupling and other efficiency overheads that reduce its throughput. In PVC, more nodes increase the likelihood of contention which can cause preemptions and reduce throughput as a consequence.

5.3 Methodology

We use a custom cycle-precise simulator to evaluate three QOS schemes – WFQ, GSF, and PVC – on performance and fairness using the metrics from Section 2.3.4. As a baseline, we use a generic NOC with no QOS support. Details of the simulation infrastructure are summarized in Table 5.2.

Experiments: To evaluate the ability of different schemes to meet fairness guarantees while maximizing throughput, we use *hotspot* and *uniform random* synthetic traffic patterns whose network behavior is easy to understand, simplifying analysis. The *uniform random* pattern is also used to understand how well the different approaches scale when the network size is increased from 64 to 256 nodes.

Additionally, we assess the ability of GSF and PVC, the two schemes without per-flow buffering, to provide efficient fine-grained communication and performance isolation in the face of a denial-of-service attack. For this experiment, we dynamically combine traffic from PARSEC [8] application traces with synthetic "attack" traffic. The traces were collected using the M5 full-system simulator [9] executing PARSEC benchmarks in their entirety. We simulate the six applications in Table 5.2, representative of different types and granularities of parallelism.

We also demonstrate PVC's ability to provide differentiated services by specifying a custom bandwidth allocation on a hotspot traffic pattern. Finally, we evaluate energy and storage overheads of different schemes. For energy analysis, we use modified versions of CACTI 6 [51] and ORION 2 [37].

For all configurations except PVC's *differentiated services* experiment, we assume that the actual traffic pattern is not known ahead of time and allocate all flows an equal share of network bandwidth.

WFQ configuration: Weighted Fair Queueing represents our ideal QOS solution with respect to fairness, performance isolation, and bandwidth utilization efficiency. Although we believe that WFQ is a poor fit for most NOC substrates due its high buffer requirements and complex schedule computation,

Network	64 and 256 nodes, 16 B link width, XY-DOR routing
Synthetic benchmarks	*hotspot* and *uniform random*; 1- and 4-flit packets, stochastically generated
PARSEC traces	*blackscholes, bodytrack, ferret, fluidanimate,* *vips, x264*: sim-medium datasets
Baseline network	6 VCs per network port, 5 flits per VC; 1 injection VC, 2 ejection VCs
WFQ network	Per-flow queueing at each router node: 64 (256) queues, 5 flits per queue
GSF network	2K (8K) cycles frame duration, 6 (24) frames in-flight, 8 cycle frame reclamation delay; 6 VCs per network port: 1 VC reserved, 5 flits/VC; 1 injection VC, 2 ejection VCs
PVC network	50K cycles frame duration, 30 (60) flit source window 6 VCs per network port: 1 VC reserved, 5 flits/VC; 1 injection VC, 2 ejection VCs

Table 5.2: Simulation methodology details; 64-node (256-node) network.

we use it as a yard-stick for evaluating the two other QOS schemes. We idealize the WFQ routers by endowing them with an unrealistically low 3-cycle pipeline latency in the contention-free case – the same latency enjoyed by GSF and PVC routers that have simple schedule computation and no per-flow queueing.

GSF configuration: The baseline GSF configuration in the 64-node network features a 2000-cycle frame, 6 in-flight frames and an 8-cycle frame reclamation delay. The routers have 6 VCs per input port, with one reserved VC for the head frame. This configuration is similar to the default setup in the original paper by Lee et al. [45], except that we use a shorter frame reclamation delay and larger frame size, both of which improve GSF's performance. For the scalability experiment, we quadruple both the frame and window size to 8000 cycles/frame and 24 frames, ensuring good performance (as shown in Figure 5.2).

PVC configuration: In a PVC network, the choice of the frame size has important implications for both throughput and fairness. Longer frames are desirable to amortize various protocol overheads and minimize the effect of gently relaxed fairness settings. On the other hand, longer frames may result in greater drift among the different flows' bandwidth consumption, increasing the likelihood of preemption for flows with high bandwidth utilization. Empirically, we found 50,000 cycles to be a good frame length for balancing these conflicting requirements. We compute each flow's reserved bandwidth quota by multiplying its rate by 95% of the frame size. Five percent of frame bandwidth is uncommitted, allowing PVC to tolerate various overheads, such as router delays and ACK return latencies, without compromising bandwidth guarantees.

Our PVC baseline is configured to maximize fairness, potentially at the expense of throughput, using unmasked bandwidth counter values for priority computation. We also show the effect of relaxed fairness settings on select experiments by increasing the bandwidth counter coarsening mask to 8 and 16 bits. The latter configuration completely eliminates all preemptions by effectively masking out the full value of the bandwidth counter.

PVC's router configuration is similar to that of GSF with 6 VCs per port, including one for reserved flits. Unlike GSF, PVC does not require a reserved VC, since preemption guarantees freedom from priority inversion. However, we found that reserving a VC can eliminate some preemptions, reducing energy and latency cost of retransmissions. PVC uses 30-flit source windows for buffering outstanding packets for possible retransmission. In the 256-node network, we double the source window to 60 flits.

For the ACK network, we assume a simple design with single-flit messages and a single 10-flit buffer per input port. Message size is 16 bits in the 64 node network (20 bits with 256 nodes), which is sufficient to cover the address, index of the acknowledged packet, hop count to the preemption point (if applicable), and status (ACK or NACK).

5.4 Evaluation

5.4.1 Quality-of-Service

First, we evaluate the QOS schemes on their ability to provide fair bandwidth allocation in a highly congested network and compare them to a system without QOS support. To do so, we use a *hotspot* traffic pattern in which all nodes send traffic to a common destination. We designate a corner node as the hotspot, and simulate 5 million cycles after the warm-up interval. Per Section 2.3.4, we are interested in *relative throughput* of different nodes. A tight distribution of throughput values across all flows is desirable, indicating a fair allocation of bandwidth to all nodes.

For each configuration, Table 5.3 shows the minimum and maximum throughput across all flows, as well as the standard deviation, all measured as a percentage of the mean throughput. We also include aggregate system throughput relative to the theoretical maximum in the measurement interval in order to assess the efficiency of different schemes in utilizing the available bandwidth.

In general, we see that all three QOS schemes are capable of fair bandwidth allocation. WFQ achieves the tightest distribution of bandwidth to nodes, benefiting from per-flow queueing and a sophisticated scheduling policy. GSF also performs very well, as source-based bandwidth reservation ensures equitable bandwidth allocation within each frame. However, GSF has the lowest aggregate throughput of any scheme, exposing inefficiencies in its bandwidth allocation. PVC has the most slack in its bandwidth distribution, but still offers good fairness with little deviation among nodes and standard deviation of just 0.8% of the mean throughput. Finally, a network with no QOS support offers high aggregate throughput but no fairness, with the node farthest from the hotspot receiving just 2.1% of the mean bandwidth.

Slack in PVC's throughput fairness has two primary causes. The first is due to fixed frame length, which allows some flows to be slightly ahead of their peers in bandwidth consumption by frame rollover. This favors nodes

	throughput (% of max)	min (% of mean)	max (% of mean)	std dev (% of mean)
No QOS	100%	2.1%	127.2%	45.7%
WFQ	100%	100.0%	100.0%	0.01%
GSF	95.3%	99.8%	100.2%	0.07%
PVC	98.3%	98.7%	101.7%	0.78%

Table 5.3: *Relative throughput* of different QOS schemes.

	mean (cycles)	max (cycles)	std dev
No QOS	264	20,675	214
WFQ	63	63	0
GSF	63	1,949	239
PVC	63	1,645	30

Table 5.4: Packet delay variation (jitter) of different QOS schemes.

closer to the hotspot, as flits from different nodes progress in wavefronts under this traffic pattern. We attribute the second source of diminished fairness to our definition of priority inversion, described in Section 5.2.2, which inhibits preemptions whenever a downstream VC is held by a packet of same or higher priority as that of a requester upstream. Thus, multiple packets of lower priority can occupy other VCs at a given downstream port and make progress whenever the VC held by the higher priority packet experiences a stall.

We also measure the packet delay variation, or jitter, associated with different QOS approaches. We modify our experimental setup to generate only single-flit packets, thus simplifying analysis. During the measurement phase, we compute the delay difference for each pair of consecutive packets within a flow. We record all such differences, and use them to compute the metrics for each flow. The aggregate mean, max and standard deviation across *all* flows is presented in Table 5.4.

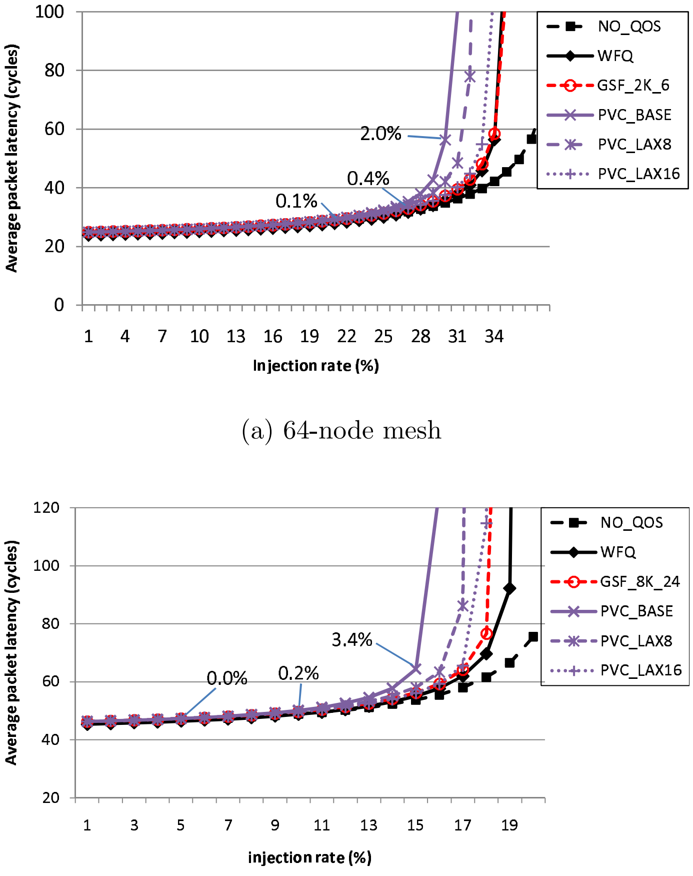
As expected, WFQ has the tightest distribution of jitter values, with virtually no variation across the flows or within any flow, benefiting from per-flow queueing coupled with a powerful scheduling function. GSF, on the other hand, shows the worst distribution of jitter values among QOS schemes due to unordered packet service within a frame. In contrast, PVC's standard deviation of jitter values is nearly eight times lower than GSF's, due to PVC's rate-based scheduling within a frame. Like GSF, PVC does not provide any jitter guarantees, as it is ultimately a frame-based approach. However, PVC's rate-based features can reduce packet delay variation in many cases, as this example shows.

5.4.2 Throughput and Performance Scalability

We use a *uniform random* traffic pattern to assess the performance of the different QOS approaches in terms of latency and maximum throughput. This all-to-all workload is self-balancing, loading all bisection links uniformly and not favoring any particular node. In fact, no fairness mechanism is necessary to achieve equal bandwidth distribution among the network nodes. Thus, this pattern is effective at exposing the performance overheads associated with the respective QOS approaches.

Figure 5.5(a) shows the latency-throughput curves for the various schemes. Three PVC curves show the difference in throughput between our baseline (conservative) fairness setting and two relaxed configurations that mask eight bits (PVC_LAX8) and the full 16 bits (PVC_LAX16) of the bandwidth counters when computing packet priorities. Labels on the baseline PVC curve show the number of wasted hops due to dropped flits as a percentage of all hop traversals at 20%, 25%, and 30% injection rates. The drop rate peaks at 35% injection rate with 5.9% of all hop traversals resulting in a preemption (not shown in the figure).

The best throughput is achieved by the generic NOC due to high VC buffer utilization. In comparison, our WFQ implementation binds each flow to a dedicated queue, causing head-of-line blocking within each flow with a deleterious effect on throughput. GSF and the most lax PVC configuration



(a) 64-node mesh

(b) 256-node mesh

Figure 5.5: Performance of WFQ, GSF and PVC on uniform random traffic. Labels on the PVC_BASE curve show the number of retried hops as a percentage of total hop traversals.

(PVC_LAX16) have similar performance, but fall short of a QOS-oblivious network on throughput due to restrictions on VC utilization. In both of these schemes, multiple packets are not allowed to share a given virtual channel to avoid priority inversion. The NO_QOS configuration is not hampered by this restriction, allowing multiple packets to queue up behind each other in a VC, thereby improving buffer utilization and boosting throughput. The PVC network with the strictest fairness setting (PVC_BASE) degrades throughput by 10% relative to the laxest configuration (PVC_LAX16) due to preemptions.

Figure 5.5(b) shows the effects of scaling the network size to 256 nodes. The relative performance of different schemes remains unchanged. The fairest PVC configuration again exhibits some throughput loss due to dropped packets, which result in 3.4% of hops wasted at a 15% injection rate and saturate near 30% injection rate (not shown in figure) with 9.5% of all hop traversals leading to a preemption. One way to combat the performance overhead of packet drop is through relaxed fairness settings, which the figure confirms to be an effective way to improve throughput.

5.4.3 Performance Isolation

To test the ability of NOC QOS schemes to provide performance isolation without per-flow queueing, we orchestrate a denial of service (DOS) attack against multi-threaded applications from the PARSEC suite. Figure 5.6 shows the configuration for this experiment. Black nodes in the left-most column are "aggressors" which send packets to the striped node in the lower-right corner of the mesh at an average rate of 20%. The rest of the nodes, including the striped node, belong to PARSEC threads. The aggressors may be a virus intentionally trying to disrupt network performance or may be benign threads accessing a shared cache bank at the noted location. We compare the average latency of PARSEC packets in this configuration to their latency executing alone on a substrate without any interference.

Our PVC baseline maps each core to a different flow with a distinct bandwidth allocation. However, PVC offers the capability to map all threads of an application to a common flow, allowing idle bandwidth from one application thread to be transparently used by another. This feature maximizes bandwidth utilization by reducing the likelihood of preemption among communication-intensive threads from the same application. To evaluate the performance of PVC that maps all PARSEC threads to a single flow, we provisioned the flow

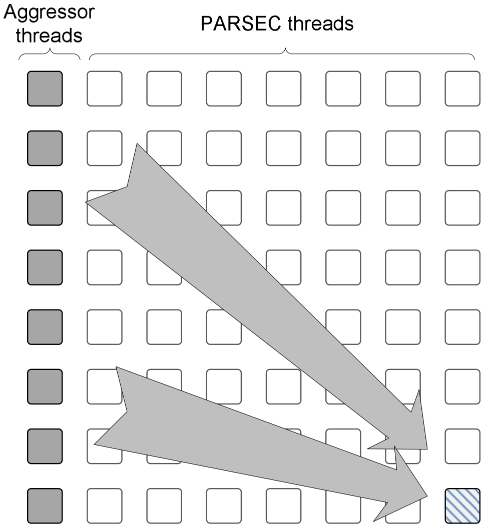


Figure 5.6: Experimental setup for PARSEC workloads.

with 7/8-ths (87.5%) of the network capacity, which is the sum of rates of individual PARSEC threads in our PVC baseline.

The results of the evaluation are presented in Figure 5.7. Five bars for each of the benchmarks show the average latency of PARSEC packets. The first bar corresponds to a network with no QOS support; the second and third are for GSF and PVC baselines, respectively; the fourth bar shows the PVC configuration with PARSEC threads aggregated into a single flow; the last bar marks the performance of each PARSEC application executing with no attack traffic.

Without QOS support, “aggressor” threads overwhelm network's limited buffering, effectively preventing PARSEC packets from entering the network. The rate at which PARSEC packets are able to acquire network resources is lower than their injection rate; as a result, their delays grow very large due to our open-loop simulation methodology.

By comparison, both GSF and PVC offer some degree of performance isolation. In a PVC network, the maximum latency increase for an average PARSEC packet over an isolated execution is 22%, with a mean increase of

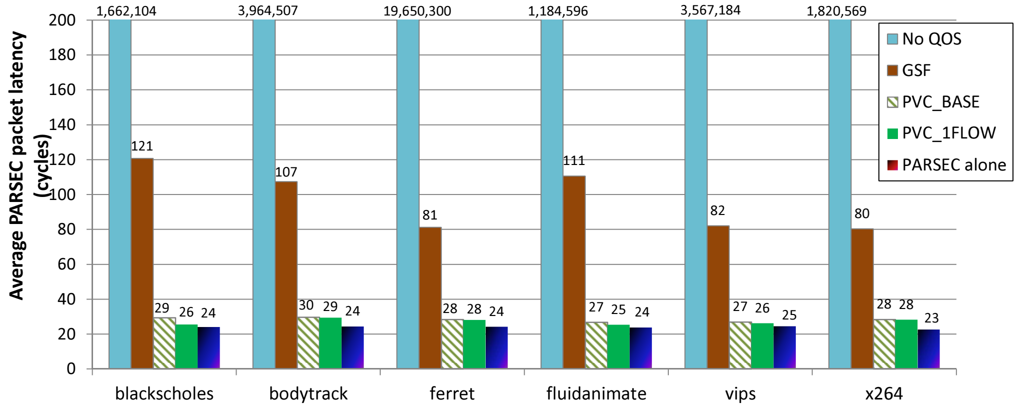


Figure 5.7: Experimental results on PARSEC workloads.

18% across the six workloads. In contrast, GSF increases average packet latency by over 500% in the worst case, with a mean of 405%. The reason for GSF's poor performance is its scheduling mechanism. Because GSF does not allow injection into the head (oldest) frame to accelerate frame reclamation, new packets are assigned to a future frame. This forces newly generated PARSEC packets to compete for buffer space with packets from aggressor threads that may belong to a more future frame, exposing PARSEC traffic to priority inversion. Importantly, GSF violates property (f) from Section 5.1.1, which states that delay should be proportional to bandwidth usage and explains GSF's poor performance in this scenario.

Finally, we note that the PVC configuration which aggregates all PARSEC threads into one flow (PVC_1FLOW) shows even better resilience to the attack than the PVC baseline, increasing PARSEC's average packet latency by just 13% over stand-alone execution. The improvement comes as a result of improved bandwidth utilization among PARSEC threads, as bandwidth reserved for threads that rarely communicate can be recycled among remaining threads.

	mean	min	max	std. dev.
1% allocation	1.01%	0.99%	1.04%	1.3%
10% allocation	10.05%	9.39%	10.28%	1.6%

Table 5.5: Differential bandwidth allocation in PVC.

5.4.4 Differentiated Services

To better support concurrent execution of multiple applications on a single substrate, PVC allows for differential bandwidth allocation to satisfy applications' diverse run-time requirements. To demonstrate PVC's ability to enforce a differential bandwidth allocation, we modify our *hotspot* configuration by provisioning four network nodes with 10% of the bandwidth each. These well-provisioned nodes are the three corners other than the hotspot, as well as a node in the center of the network. The rest of the nodes each get 1% of the bandwidth. The packet generators at the nodes exceed the provisioned rate, ensuring the relevance of the QOS mechanism.

Table 5.5 shows the mean, minimum, and maximum throughput among the nodes, along with the standard deviation from the mean, for the two allocations. PVC is successful in differentiated bandwidth provisioning with a standard deviation of under 2% for both allocations. The difference between minimum and maximum throughput among the nodes with a small bandwidth allocation is 4.5%, and 3.5% among those with a large allocation. The greater difference in bandwidth distribution among the nodes with a 1% allocation is partly due to preemptions suffered by certain nodes in the path of flows with high provisioned bandwidth. The fewer hops a flow with the low allocation shares with a high-allocation flow, the less likely it is to experience preemptions and diminished throughput.

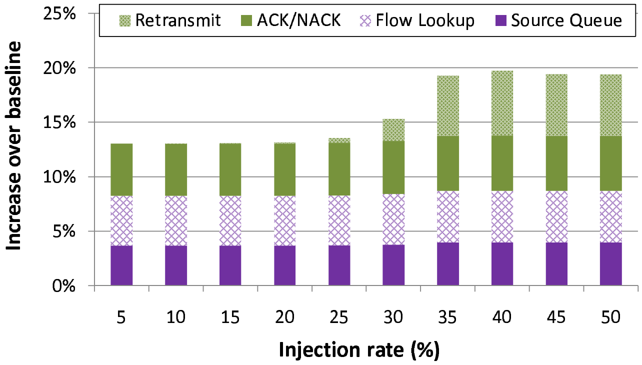


Figure 5.8: PVC energy overhead over a generic NOC with no QOS support.

5.4.5 Energy

Figure 5.8 shows the energy expended in a 64-node PVC network relative to a baseline NOC with no QOS support on a *uniform random* traffic pattern. Four primary components of PVC's energy overhead are the source buffers, flow table lookups, ACK network, and retransmission of preempted messages.

Prior to saturation, PVC expends 13% more energy than the baseline due to source queue writes, flow table look-ups and updates, and ACK network overhead. As few preemptions occur before saturation, retransmissions incur very little energy overhead. As discussed in Section 5.4.2, the preemption rate peaks when the injection rate reaches 35% and holds steads thereafter, which Figure 5.8 confirms. In saturation, retransmissions are responsible for an additional 6% of the energy consumed. Other components of PVC's energy overhead also increase by 5-8% in saturation, contributing an insignificant amount to the overall energy budget.

WFQ, GSF, and PVC each have energy advantages and disadvantages. WFQ requires large per-flow buffers within each router, and a message must be written into and read from each of these as it traverses the network. GSF eliminates these buffers, but instead requires large source queues. Additionally, the large buffer capacity in both WFQ and GSF incur a non-trivial leakage energy penalty. PVC requires only small source buffers and also eliminates

	64 nodes		**256 nodes**	
	bytes	**relative**	**bytes**	**relative**
No QOS	1,920	1	1,920	1
WFQ	5,120	2.7	20,480	10.7
GSF	33,920	17.7	129,920	67.7
PVC	3,376	1.8	6,564	3.4

Table 5.6: Per-node storage requirements of different QOS schemes. Absolute values and relative to a generic NOC without QOS.

the per-flow buffers, giving it a potential storage energy advantage relative to the other two schemes.

5.4.6 Storage Requirements

We compare the storage requirements of different QOS schemes in 64- and 256-node networks in Table 5.6. For each configuration, both the absolute amount of storage, in bytes, and relative increase over a generic baseline with no QOS support is specified. For simplicity, we ignore the area overhead of the packet scheduling and buffer management logic, as well as the buffering at the local interfaces.

In WFQ, the primary source of storage overhead are the per-flow queues at each routing node. In contrast, GSF does not require per-flow buffering at the routers, instead necessitating large queues at the source nodes. PVC has three primary sources of area overhead: per-flow state in each router, buffering for outstanding transactions at each source interface, and flit buffers in the ACK network.

To store per-flow state, PVC needs bandwidth counters (one per flow) for each output port, as well as a reserved rate register and a reserved bandwidth register that may be shared across the ports, for a total of seven registers per flow. With a frame duration of 50,000 cycles or less, PVC requires 16 bits of storage per register.

In the 64-node network, PVC has 1.5 times less buffering than WFQ and 10 times less than GSF. In the larger network, PVC's storage footprint is 3 times smaller than WFQ's and 20 times smaller than GSF. Although the difference between WFQ and PVC may not appear significant, WFQ's scheduling and buffering overheads are in the critical path of each router node, which is undesirable in latency and energy sensitive on-chip interconnects.

5.5 Discussion

The preemptive aspect of PVC enables a low-cost NOC QOS architecture with strong guarantees. Unfortunately, preemptions tend to diminish network throughput and energy-efficiency. Because preemptions arise as a result of limited network storage, we wish to study the effect of flow control, which manages storage resources, on fairness and preemption incidence. We also study the role of topology in a PVC network. More specifically, we wish to understand whether reducing the network diameter helps lower the preemption rate by reducing the number of arbitration events.

5.5.1 NOC Flow Control Mechanisms

Flow control mechanisms manage two essential network resources: buffers and bandwidth [15]. In general, flow control schemes can be classified into one of two categories: packet-level and flit-level. As the name implies, packet-level approaches manage network resources at a packet granularity. As such, they allocate storage at each node for a whole packet and switch an entire packet completely before transferring another packet via the same interface. Examples of packet-level flow control approaches include store-and-forward and virtual cut-through (VCT) [38]. In contrast, flit-level mechanisms assign network buffers and bandwidth on a per-flit basis. Best-known examples of such schemes include wormhole [12] and virtual channel [11] flow control.

By managing buffers and bandwidth at a fine granularity, flit-level flow control approaches can reduce buffer requirements compared to packet-level regimes. In fact, most NOC designs are based on flit-level wormhole or virtual channel flow control. However, modern NOCs can be designed with packet-level flow control with no loss in efficiency, especially in cases where the baseline design is based on virtual-channel flow control. The reason is that wide NOC datapaths translate even the largest packets into a small number of flits. For instance, a 576-bit packet that encapsulates a 64-byte cache line requires just five flits to transfer over a 128-bit link. In a mesh network with a 3-cycle router pipeline and one cycle of wire delay between adjacent routers, the minimum amount of buffering to cover the credit round-trip time is at least five flits. Thus, an entire 5-flit packet can comfortably fit in a router input buffer. Long link spans found in richly-connected topologies necessitate additional buffer capacities, a feature that further diminishes the benefit of flit-level flow control.

The PVC network evaluated in Section 5.4 was based on a flit-level virtual channel flow control architecture. In this design, flits from different packets may interleave in the channel. One problem with such interleaving is that it can increase the average packet transfer latency and retard VC turnover. Slow VC turnover in a congested network raises the likelihood of preemption. On the other hand, a packet-level architecture has the effect of accelerating VC turnover in the presence of multi-flit packets due to its interleaving-free nature. By relieving VC pressure through faster VC recycling, packet-level flow control may help in reducing preemption incidence.

Given a PVC-enabled network with packet-based flow control, it may also be possible to reduce preemption incidence by inhibiting the preemptive mechanism while a packet transfer to a requested output port is in progress. Preemptions should only be initiated at the time of VC allocation, which begins when a port is free or about to become free. However, a potential danger of such a design is that fairness may suffer if a high priority packet is delayed by the transfer of a lower-priority one.

An additional benefit of packet-level flow control is that it can be realized with a fused buffer and switch allocator. In contrast, flit-level virtual

Flow control	throughput (% of max)	min (% of mean)	max (% of mean)	std dev (% of mean)
Flit	98.3%	98.7%	101.7%	0.78%
Packet	100%	99.0%	101.6%	0.68%

Table 5.7: *Relative throughput* under flit- and packet-level flow control in a 64-terminal PVC network.

channel architectures necessitate separate VC and switch allocators that decouple virtual per-packet storage (VC) from physical flit buffers. NOC routers implementing packet-level allocators can thus eliminate an entire allocation stage thereby reducing node delay.

5.5.2 Effect of Flow Control on Fairness

We start by comparing the fairness of flit- and packet-level allocators in a PVC-enabled mesh network. Our experimental setup is identical to the one in Section 5.4.1 and is based on a hotspot traffic pattern in a 64-terminal mesh network. The baseline router uses flit-level allocation with a 3-stage router pipeline previously evaluated. We compare it against a VCT-like architecture that features a 2-stage pipeline with a fused VC-switch allocator operating at a packet granularity. As described above, the preemptive mechanism in a router using packet-level allocation is inhibited during transfers to the requested output.

Table 5.7 summarizes the results of the evaluation. Compared to flit-level flow control, a packet-level architecture does not diminish fairness. In fact, the packet-granularity allocator improves the standard deviation from the mean by nearly 13%, which indicates a tighter bandwidth distribution among the nodes. In addition, packet-level flow control improves overall network throughput (second column in the table) by virtue of faster VC turnover and a shallower router pipeline.

5.5.3 Effect of Flow Control and Topology on Preemption Incidence

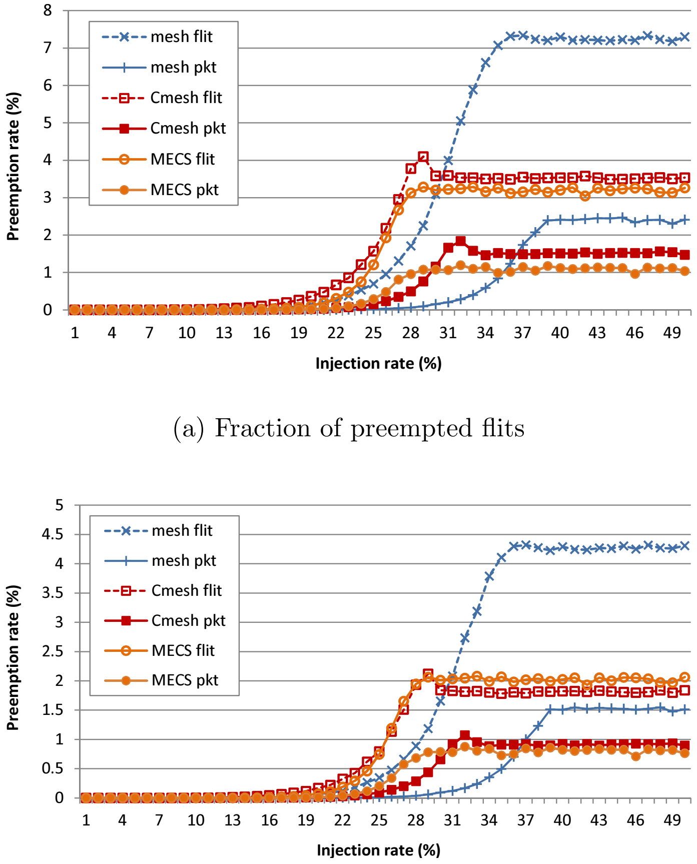
We examine PVC network efficiency under different flow control regimes as well as topologies. In this context, network efficiency refers to susceptibility of the network to preemptions. Since preemptions reduce network throughput and increase energy consumption, a network that reduces preemption incidence is more efficient than a preemption-prone system.

Per Section 5.5.1, we anticipate that packet-granularity flow control with the optimized preemptive mechanism that inhibits preemptions to active output ports will reduce the incidence of packet drop compared to the baseline flit-level allocator. With respect to topologies, we hypothesize that organizations that improve network connectivity are less susceptible to preemptions than architectures with a large network diameter. This hypothesis is based on the observation that preemptions are triggered by resource allocation events, which only occur at router traversals. Thus, an organization that diminishes the network diameter also reduces the number of arbitration points, which, in turn, cuts down the likelihood of preemption.

To evaluate our hypotheses, we compare preemption incidence in a mesh, concentrated mesh (Cmesh), and Multidrop Express Channels (MECS) topologies on random traffic. Each topology is assessed with flit- and packet-level flow control. Our metrics are the number of preempted flits and the number of replayed hops. Since the hop count is a function of the topology, we express it in hops in an unconcentrated mesh network (mesh-hops).

64-terminal network

Figure 5.9 plots the preemption rate in (a) flits and (b) mesh-hops against network load in a 64-terminal system. For a given organization, the preemption rate in flits always exceeds the hop-level metric. The reason is that preemptions tend to occur at or close to the source node, since a packet that has advanced by several hops in a highly congested environment is likely of suffi-



(a) Fraction of preempted flits

(b) Fraction of replayed hops

Figure 5.9: Preemption incidence in mesh, Cmesh, and MECS topologies with flit- and packet-level flow control in a 64-terminal network.

ciently high priority that its chances of being preempted in subsequent hops are low. In saturation, preemption rates are steady across all simulated organizations, which allows us to conclude that PVC-based systems are stable under load. The highest preemption rates are observed in a mesh topology with flit-level flow control, with under 8% of flits (under 5% of hops) discarded.

In terms of flow control, the proposed packet-level allocator significantly reduces the maximum preemption rate compared to the flit-level architecture.

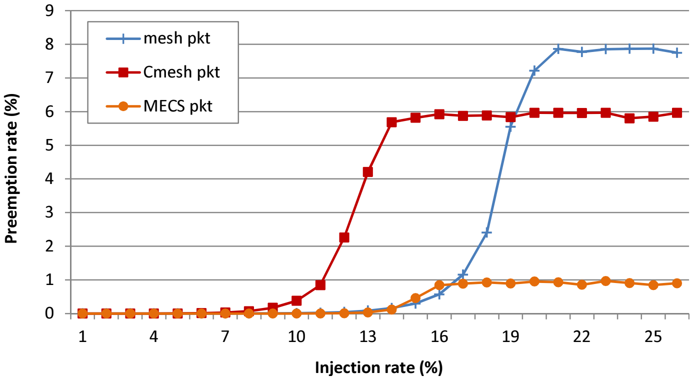
This conclusion holds for all three topologies and both preemption metrics. Thus, a mesh network with packet-granularity flow control reduces the maximum preemption incidence by a factor of three compared to a flit-level architecture. Similarly, Cmesh and MECS topologies enjoy a 2.4x and 3.1x lower preemption rate, respectively, with packet-level flow control.

Our hypothesis regarding the relationship between network diameter and susceptibility to preemptions turns out to be only partly correct. When all three topologies are in saturation, the hypothesis holds. Thus, for a given flow control regime, the mesh network is clearly more susceptible to preemptions than the Cmesh, which, in turn, is inferior to MECS.

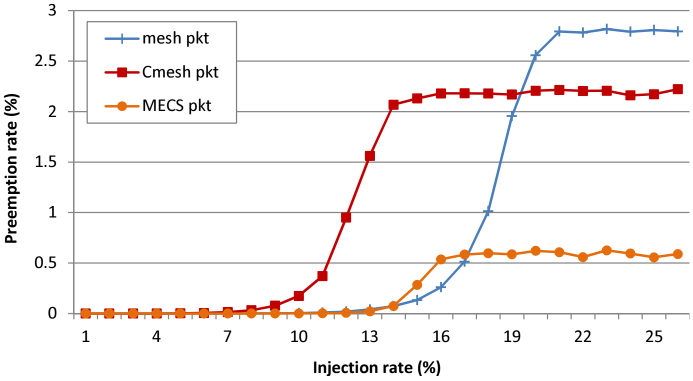
The hypothesis breaks down due to the differences in maximum sustained load across the topologies. Since the preemption rate is highest in saturation, a network in saturation may likely experience higher preemption incidence than another network with a different topology that is not in saturation under the same load and traffic parameters. From Section 4.5.1, we know that the mesh topology on random traffic has a performance advantage at high load rates over Cmesh and MECS organizations. As a result, as the latter topologies enter saturation, their preemption rate exceeds that of the mesh. The advantage of lower-diameter networks in reducing preemption incidence emerges only once the mesh approaches saturation.

256-terminal network

Figure 5.10 shows the effect of scaling the network size to 256 nodes. To focus the discussion, we only present the results for packet-level flow control due to its greater efficiency compared to flit-level allocation. The larger network size amplifies the differences between the topologies. In particular, the low-diameter MECS organization shows significantly lower susceptibility to preemptions than mesh-based designs. In mesh networks, the increased network diameter in the 256-terminal organization results in more arbitration points compared to the 64-terminal system, which increases the probability of discard. In contrast, the network diameter of MECS stays remains unchanged



(a) Fraction of preempted flits



(b) Fraction of replayed hops

Figure 5.10: Preemption incidence in mesh, Cmesh, and MECS topologies with packet-level flow control in a 256-terminal network.

under scaling, reducing the effect of network size on preemption rate. As a result, preemption incidence in MECS at high load rates is 4-8 times lower than that of mesh-based NOCs.

5.6 Conclusion

Future CMP and SOC substrates will integrate hundreds or thousands of compute and memory elements on a single die. These elements will be connected by an on-chip network, which will shoulder the responsibility of providing fair access to shared resources while meeting performance, area, and energy targets. Prior network QOS schemes suffer from high buffer overheads, complex scheduling functions or poor bandwidth utilization, motivating us to propose Preemptive Virtual Clock, a novel QOS scheme specifically designed for on-chip interconnects. By combining features of frame-based and rate-based approaches, PVC provides strong guarantees, enforces flow isolation, and enables efficient bandwidth utilization with modest hardware cost and complexity. PVC does not require per-flow buffering, reducing router area and energy footprint. Priority inversion in a PVC network is averted through preemption of lower-priority packets. To ensure packet delivery in a discarding NOC, PVC relies on a dedicated low-bandwidth ACK network and a small window of outstanding transactions at each node. Finally, PVC enables flexibility in network provisioning by allowing bandwidth to be allocated at any granularity from a single thread to an application to a user.

An evaluation of PVC shows that it can guarantee fairness and provide differentiated services with low latency and good throughput. PVC also delivers strong performance isolation, demonstrated in a denial-of-service scenario against several PARSEC benchmarks. Results confirm that the average latency of PARSEC packets increases by less than 22% with PVC over their execution in isolation. In comparison, a previously proposed NOC QOS scheme called GSF causes latency to increase by up to 500%.

The preemptive QOS architecture minimizes PVC's storage requirements, which helps keep the NOC area footprint small. While limited buffering at router input ports is desirable from an energy-efficiency standpoint, it can result in preemptions at high load rates when buffers are scare. Preemptions decrease network energy-efficiency due to the need to retransmit discarded packets. Our work shows that various mechanisms can successfully reduce preemption incidence in a PVC network, thereby boosting efficiency and performance. These mechanisms include coarsening of priority levels, which has a side-effect of weakening the guarantees, as well as using packet-level, instead of flit-level, flow control. We expect that additional research aimed at reducing preemption incidence will further reduce the overheads of PVC.

Chapter 6

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

In this chapter, we focus on NOC scalability from the perspective of energy, area, performance, and quality-of-service. While Chapter 4 showed that a direct low-diameter topology improves latency and energy efficiency in NOCs with dozens of nodes, we identify critical scalability bottlenecks in such 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 kilo-terminal NOC with a low-diameter MECS topology and PVC QOS support may require 750 VCs per router and over 12 MBs of buffering per chip, as detailed in Sec. 6.2.1.

Portions of this chapter appear in the published version of the work [33].

To tackle the scalability challenges of existing NOCs, we propose a hybrid network-on-chip 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 [48]. 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 QOS. Network energy efficiency is 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 with a 125 W power budget, the proposed NOC consumes under 7% of the overall area and 19% of power at a sustained network load of 10%.

The remainder of this chapter is structured as follows. Section 6.1 examines existing NOC technologies and their scalability bottlenecks. Section 6.2 describes the proposed kilo-NOC architecture. Sections 6.3 and 6.4 present the evaluation methodology and results. Finally, Section 6.5 concludes this chapter.

6.1 NOC Scalability Bottlenecks

This section examines existing NOC technologies for kilo-terminal chips and analyzes their scalability bottlenecks. We start with conventional NOC attributes – topology, flow control, and routing – followed by quality-of-service technologies.

Topology

Low-diameter topologies are a critical kilo-NOC technology, since they reduce the significant delay and energy costs of router traversals in high-radix networks. 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 6.1. The flattened butterfly requires O(k^2) bisection channels per row/column, where k is the network radix, to support

	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	$log(4v)$	$log(k \cdot v)$	$log(k \cdot v)$

Table 6.1: Scalability of NOC topologies. k: network radix, v: per-port VC count, C: a small integer.

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. Thus, 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. Per-port buffer requirements grow to cover the longer wire flights times, caused by increased resistivity of wires under technology scaling. The relationship between network size and router buffer requirements under technology scaling 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, as shown in Figure 4.1(c). While switch complexity is not a concern in MECS, asymmetry in the number of input and output ports can limit throughput in this topology.

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.

Flow Control

In a Kilo-NOC with a low-diameter topology, long channel traversal times necessitate deep buffers to cover the round-trip credit latency. Meanwhile, wide channels keep the number of flits per network packet small. These two attributes diminish the benefits of flit-level flow control traditionally used in NOCs, since routers typically have enough buffer capacity for multiple packets. Compared to flit-level flow control, a packet-level architecture couples bandwidth and storage allocation, which has a desirable effect of reducing the number of arbitration stages. Coarser flow control granularity also helps amortize 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. The goal is to reduce router complexity by distributing the buffering and flow control logic. To that end, 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 [43]. An alternative proposal by Michelogiannakis et al. advocates a pure elastic-buffered architecture without any virtual channels [48].

An important challenge for existing elastic buffer architectures is deadlock avoidance. 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 VC requirements in a low-diameter NOC under technology scaling grow quadratically with network radix, as explained above, impeding scalability. On the other hand, a pure elastic-buffered architecture enjoys linear scaling in router storage requirements, but needs multiple networks for deadlock avoidance (one network for each packet class), incurring chip area and wiring expense.

Routing

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 [40, 31]; 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.

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. We therefore 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.

A quality-of-service architecture based on Preemptive Virtual Clock, introduced in Chapter 5, significantly reduces the cost of providing QOS support as compared to prior work. However, in a low-diameter topology, PVC's virtual channel requirements grow quadratically with network radix (the analysis is similar to the one under Topology, above). PVC necessitates multiple virtual channels because packets from different flows are not allowed to share a VC to prevent priority inversion within a FIFO buffer. Thus, longer links require more, 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 in kilo-node chips.

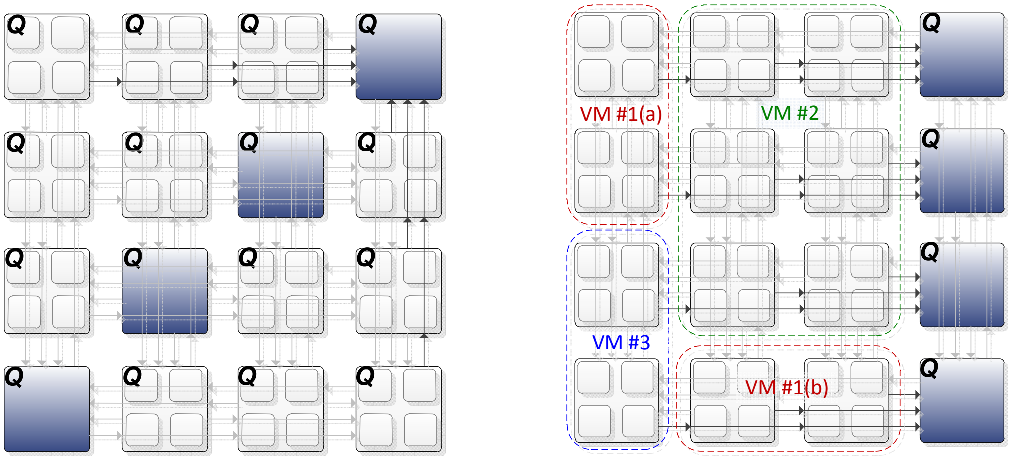
Summary

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. Our analysis identifies buffer requirements of low-diameter networks as a true scalability bottleneck. Buffer demands in such networks grow quadratically with network radix under technology scaling, diminishing area- and energy-efficiency of large-scale NOCs. Quality-of-service further increases storage demands and creates additional overheads. Supporting tomorrow's Kilo-NOC configurations requires addressing these scalability bottlenecks.

6.2 Kilo-NOC Architecture

6.2.1 Baseline Design

Our target in this work is a 1024-tile CMP in 15 nm technology. Figure 6.1(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 [5] to reduce the number of network nodes to 256 by integrating four terminals at a single router via a fast crossbar switch. The nodes are interconnected via a richly connected



(a) Baseline QOS-enabled CMP

(b) CMP with topology-aware QOS

Figure 6.1: 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.

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. Without loss of generality, we assume that QOS is used to provide isolation among VMs. However, the following discussion equally applies to application-level service guarantees.

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} [15]:

$$t_{RTCT} = 2t_{wire} + t_{flit} + t_{credit} + 1 \tag{6.1}$$

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).

6.2.2 Topology-aware QOS Architecture

Our first optimization target is the QOS mechanism. As noted in Section 6.1, 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 6.1(b) shows the proposed organization. While the SR column in the figure 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 6.1(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 6.1(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 [47].

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.

6.2.3 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. 6.2.1, 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 explained in Section 6.1. The resulting area and wire overheads diminish the appeal of elastic buffering.

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 architecture 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

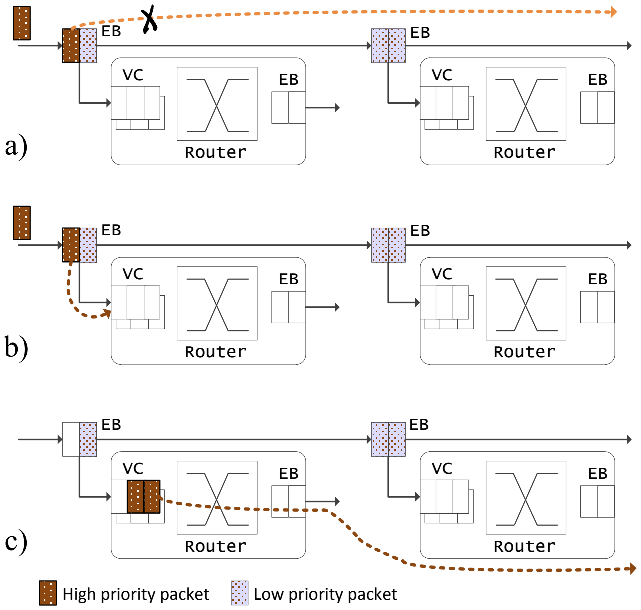


Figure 6.2: Elastic buffer deadlock avoidance.

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 6.2 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

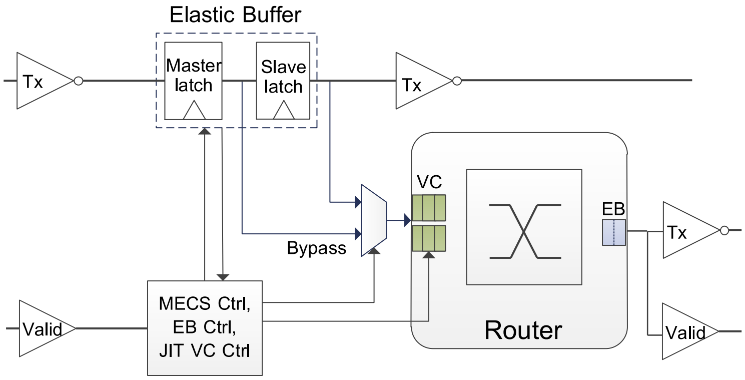


Figure 6.3: MECS with deadlock-free elastic buffer.

enforced at the end points.

Figure 6.3 shows the proposed design in the context of a MECS network. The EB, based on the design by Michelogiannakis et al. [48], 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.

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 6.3). A reply can get stuck at this EB behind a request packet, violating 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.

6.3 Experimental Methodology

Area and energy

Our target configuration is a 1024-tile (256 node) CMP in a 15 nm technology with on-chip voltage of 0.7 V. For both area and energy estimation, we use a combination of analytical models [36, 5], Orion [37], CACTI [51], previously published data [61], and logic 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. When valid is transmitted through the network, we assume an activity factor of 25% for wires and logic, which corresponds to a random bit stream.

Switch fabrics: We modify Orion to more accurately model crossbar fabrics, carefully accounting for the asymmetric switch design in MECS. Crossbar wires are routed on local metal layers with a 50 nm pitch and 2x spacing. To reduce switch energy, we split input and output crossbar wires into two segments each and activate only the necessary segments [79]. Similarly, when profitable, we segment the long wires feeding the crossbar in MECS routers with many network ports sharing a switch interface.

Buffers: We assume that VC FIFOs and PVC's flow state tables are SRAM-based. To model small SRAM FIFOs at router input ports, we modify CACTI to support the required SRAM configurations with data flow typical of a NOC router. 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.

Channels: To reduce interconnect energy, we adopt a low-swing signaling scheme over differential wires proposed by Schinkel et al. [61]. 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 improvement in area efficiency is achieved through elimination of repeaters required on full-swing links. Wire parameters are summarized in Table 6.2.

Network configurations:

Network details are summarized in Table 6.2. There are 1024 terminals in the system. We apply four-way concentration at each router to reduce the number network nodes to 256, of which 64 are 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 [32]. 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

Network	1024 terminals: 256 concentrated nodes, including 64 shared resources 128-bit links, DOR routing, packet-level flow control
Channels	Low-voltage-swing signaling over intermediate-layer wires: pitch: 100 nm, R: 8.6 kΩ/mm, C: 190 fF/mm voltage swing: 125 mV
MECS (no PVC)	Router: 2 VCs/port, 35 flits/VC, 3 stage pipeline: VA-local, VA-global, XT
MECS + PVC	Router: 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 EB router pipeline: XA, XT Within SR: MECS + PVC
K-MECS	Outside SR: single-network EB MECS 1 EB stage b/w adjacent routers JIT-VC flow control Router: 2 VC/port, 4 flits/VC, 2 stages: XA, XT Within SR: MECS + PVC
Cmesh + PVC	Router: 6 VCs/port, 4 flits/VC, 2 stage pipeline: VA, XT
common	1 injection VC, 2 ejection VCs per terminal
PVC QOS	400K cycles per frame interval
Workloads	Synthetic: *hotspot* and *uniform random*; 1,4 flits/packet PARSEC traces: see Table 6.3

Table 6.2: Simulated network characteristics.

described in Sec. 6.2.1.

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 and are identical to the MECS+PVC configuration in Table 6.2. 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 [48]. 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 [43], 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 [5] 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 [9] to collect memory access traces from a full system running PARSEC v2.1 benchmarks [8]. 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 [35], the

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

Table 6.3: Simulated PARSEC traces.

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 6.3 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).

6.4 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.

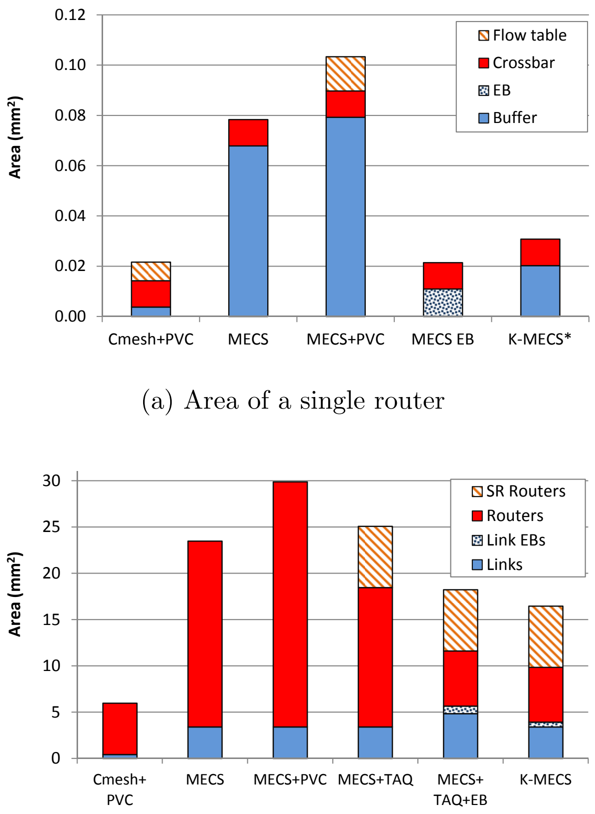
6.4.1 Area

Our router 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 6.4(a). For TAQ-enabled configurations, the area breakdown of routers inside the shared regions is reflected in the MECS+PVC bar. We denote a pure elastic buffered router outside of a shared region as MECS+EB; in contrast, the configuration labeled MECS+TAQ+EB in Table 6.2 refers to the entire heterogeneous network. Similarly, K-MECS* refers to the proposed EB-enabled router, whereas K-MECS represents an entire NOC. The intent of this additional notation is to help differentiate the different router organizations in heterogeneous NOCs.

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 6.4(b) breaks down total network area into four resource types: links, link-integrated EBs, regular routers, and SR routers. The last category is 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



(a) Area of a single router

(b) Total network area

Figure 6.4: Router and network area efficiency.

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. Normalized to the same bisection bandwidth as MECS-based topologies, the area footprint of a Cmesh network grows to 47.8 mm^2, which represents a 66% increase over K-MECS.

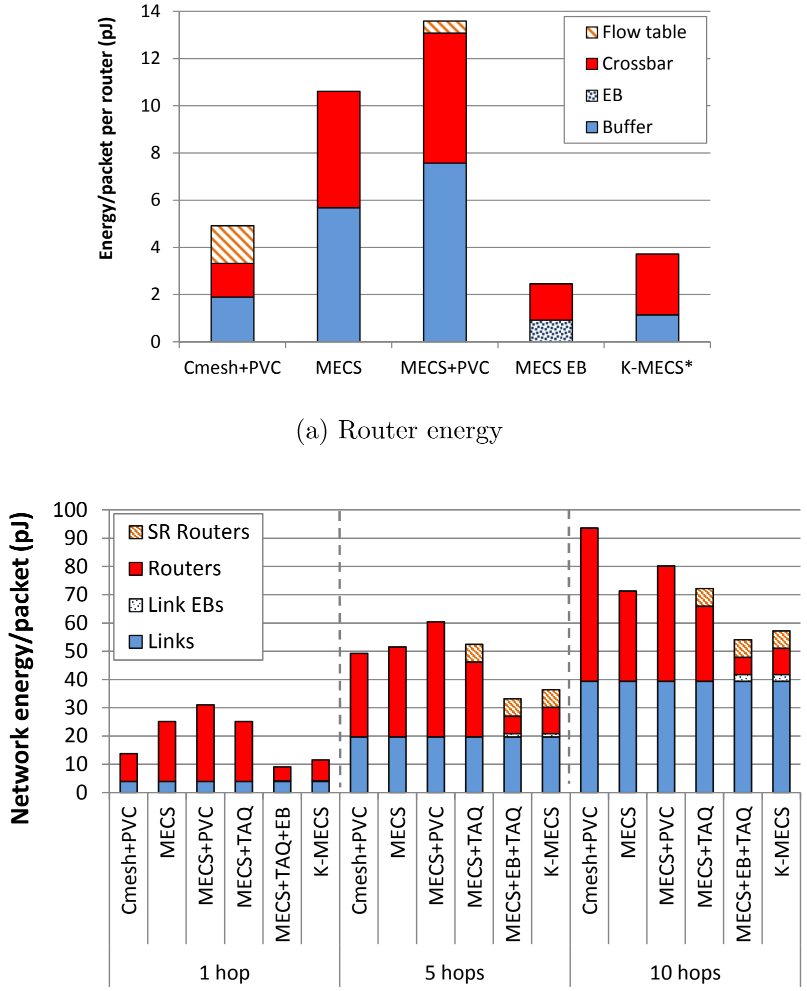
6.4.2 Energy

Figure 6.5(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. Generally, in mesh networks, 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 with deterministic routing, only a fraction of all flows can enter a router through any given port. As a result, flow state is distributed among many ports in a MECS router, which helps reduce look-up energy at each port. Although the total required per-flow storage is comparable in Cmesh and MECS, the large physical tables in a Cmesh router incur a significant per-access energy penalty.

Figure 6.5(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



(a) Router energy

(b) Network energy per packet

Figure 6.5: Router and network energy efficiency.

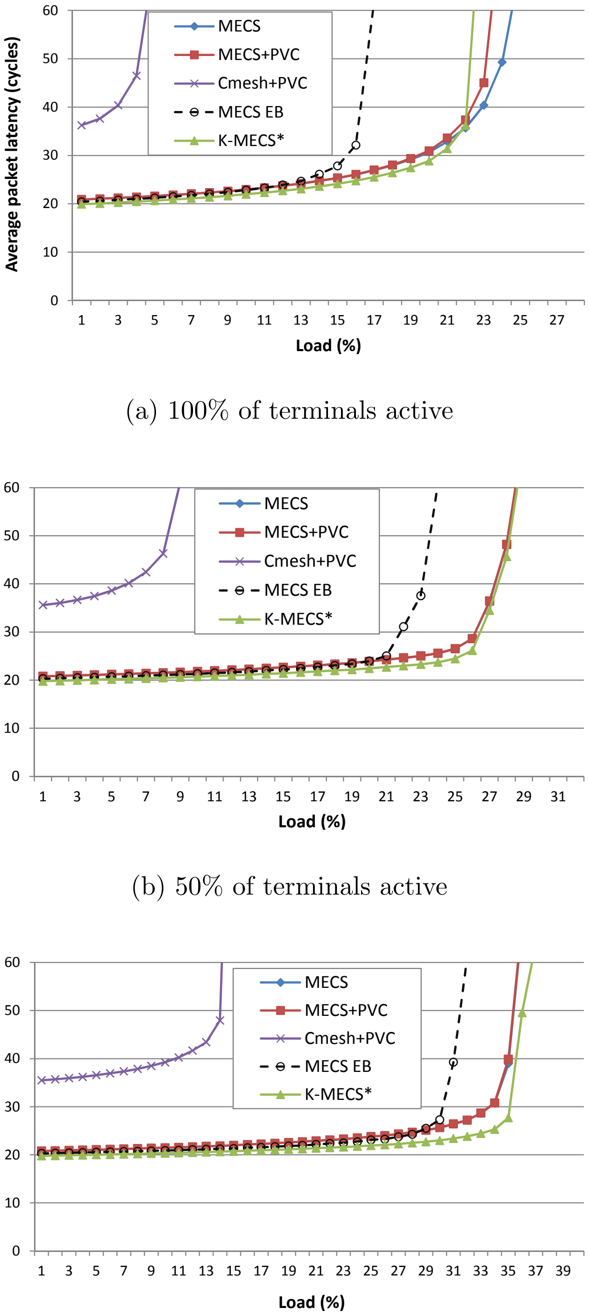
routes thanks to a reduction in buffer and switch input power. K-MECS reduces NOC energy by 16-63% over remaining network architectures on local traffic and by 20-40% on non-local patterns. Links are responsible for a significant fraction of overall energy expense, diminishing the benefits of router energy optimizations. For instance, links account for 69% of the energy expended 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.

6.4.3 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 6.6 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. As before, a different notation is meant to differentiate these EB-enabled networks from the heterogeneous NOCs of which they would be part of. MECS+EB has dedicated request/reply networks. K-MECS* uses the JIT-VC allocation mechanism described in Section 6.2.3. In networks equipped with PVC, we disable the preemptive mechanism to avoid preemption-related throughput losses.

Among the evaluated organizations, low-diameter topologies with router-side buffering offer superior throughput. With 100% of the 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



(a) 100% of terminals active

(b) 50% of terminals active

(c) 25% of terminals active

Figure 6.6: Performance comparison of different topologies on random traffic.

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 applies backpressure toward 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.

6.4.4 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 6.4 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

	min wrt mean	max wrt 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%

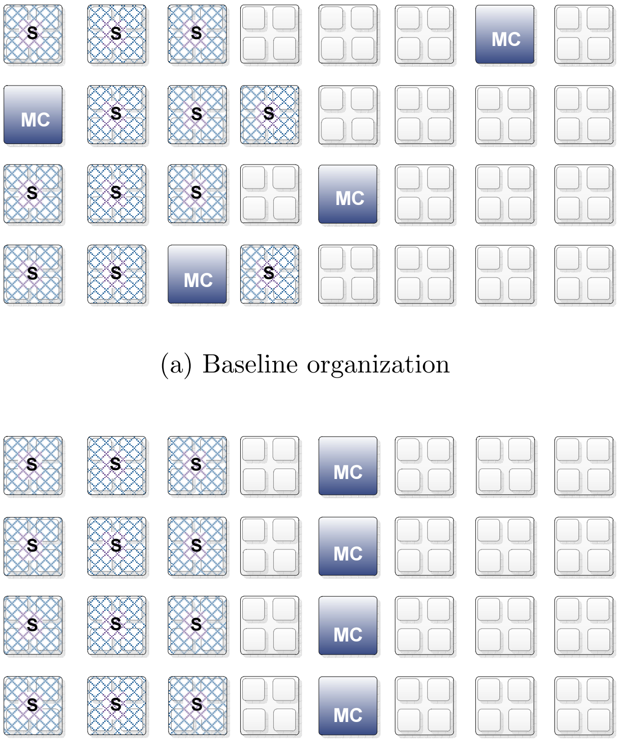
Table 6.4: Fairness and throughput of different NOCs.

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.

6.4.5 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 6.7 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



(a) Baseline organization

(b) Organization with a shared resource region

Figure 6.7: Experimental setup for PARSEC trace-based evaluation. PARSEC nodes: plain; Streaming nodes: striped; Memory controllers: shaded.

controllers. Baseline MECS and CMesh networks use a staggered memory controller placement (Figure 6.7(a)). This placement strategy is motivated by the desire to better distribute memory traffic and reduce contention at the MC interfaces, as suggested by Abts et al. [1]. The remaining NOCs employ a single shared region containing the four MC tiles (Figure 6.7(b)).

Figure 6.8 plots the slowdown of PARSEC packets in the presence of streaming traffic for the various network organizations. 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

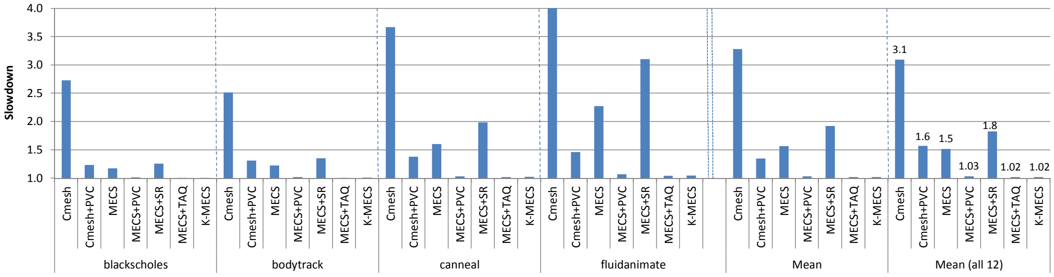


Figure 6.8: Average packet slowdown on PARSEC workloads with adversarial traffic.

provided by the shared region organization, we introduce a MECS+SR variant that is similar to the MECS+TAQ network but without any QOS hardware either inside or outside of the shared region. 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.

	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%

Table 6.5: Network area and power efficiency.

6.4.6 Summary

Table 6.5 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 6.4.2. 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.

6.5 Conclusion

In this chapter, we proposed and evaluated architectures for kiloscale networks-on-chip that address area, energy, and 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. We extend prior work on low-diameter architectures for on-chip networks [40, 31] by studying their scalability and QOS properties. 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.

Chapter 7

Conclusions

Steady improvements in semiconductor process technology have enabled single-chip systems that integrate dozens of cores, cache banks, memory controllers, and other assets. Applications and system software have evolved as well to accommodate the parallel processing capabilities of these multicore substrates. As a result, applications across a variety of market segments have become increasingly parallel, while virtualization techniques have enabled multiple operating systems to share a die. It is broadly expected that future generations of chips will be characterized by greater degrees of resource integration, thread-level parallelism, and on-die software consolidation through virtualization.

A critical resource in chip multiprocessors and systems-on-a-chip is the on-die interconnection network. Similar to the off-chip interconnect of conventional multiprocessors, the design of the NOC carries significant performance implications. From a software perspective, a low-latency NOC reduces the memory access time; in the case of parallel applications, a fast, high-bandwidth network also reduces communication and synchronizations costs.

Parallel computers have been built for a variety of purposes over a number of decades. As a result, the art and science of designing off-chip interconnects is well established and gradually evolves with technology. In contrast, the field of networks-on-chip is young and distinguished by unique constraints and demands compared to off-chip networks. NOCs are characterized by modest chip area and power budgets, short on-die communication distances, and wire routing restrictions. These features impede the use of complex router microarchitectures, low-diameter topologies, and sophisticated non-minimal routing schemes developed for off-chip networks. Existing network architectures that are easily mapped onto a die, such as rings and meshes, show poor scalability properties in terms of performance and energy-efficiency beyond a few dozen nodes. Resource-intensive features such as quality-of-service support further burden the NOC through increased cost and performance overhead. These observations motivate the design of new network architectures, designed from the ground up with die-level characteristics in mind.

7.1 Dissertation Summary

This dissertation proposes architectural and microarchitectural mechanisms for performance, efficiency, and quality-of-service in on-chip networks of highly-integrated chips. These mechanisms cover the principal aspects of interconnection networks, namely routing, topology, flow control, and QOS. The proposed solutions are complementary and together afford a NOC capable of interconnecting over a thousand nodes with low latency, modest area and energy footprint, and strong service guarantees.

7.1.1 Congestion-Aware Routing

Energy and area considerations in on-chip networks preclude the use of non-minimal adaptive routing schemes and buffer-rich routers. Without these features, existing NOCs do not reach their performance potential as evidenced by high latency and poor throughput on many workloads. Performance suffers due to congestion effects that result from simplistic routing policies and limited buffer resources.

We demonstrate that congestion-aware minimal adaptive routing is a cost-effective approach to boosting performance in resource-limited interconnects through load balancing of network traffic. By judiciously spreading traffic across network links, congestion-aware routing improves resource utilization and reduces the effect of localized hotspots on network performance. One problem with existing adaptive routers is their reliance on local-only congestion indicators. Without knowledge of global network state, such locally-adaptive routers can compromise performance on balanced traffic patterns through myopic decisions.

Together with another student, I developed Regional Congestion Awareness (RCA), a light-weight approach for informing the route selection process in adaptive routers with broader knowledge of network state. At each network router, RCA aggregates local and non-local congestion indicators and uses the information to (a) inform the local routing policy, and (b) notify adjacent routers of local and downstream congestion. By considering congestion information from certain network regions and assigning different priorities to local and non-local estimates, RCA supports a range of policies. RCA uses a dedicated low-bandwidth reduction network to propagate congestion information among interconnected nodes, an approach that scales naturally to large network configurations with minimal hardware and wiring cost.

Our evaluation reveals that on irregular traffic, RCA improves performance compared to both deterministic and locally-adaptive routing. On balanced permutations that benefit from deterministic routing, RCA preserves performance to a higher degree compared to locally-adaptive designs. For a given level of performance, RCA requires up to 50% less buffering than a locally-adaptive router, a feature that may be used to improve network area and energy efficiency. Starting with an RCA baseline tuned for a 64-node mesh, we observed greater performance gains in a small 16-node network and diminished, yet still considerable, benefit in a large 256-node NOC. Careful tuning of the congestion-awareness mechanism to accommodate network characteristics will likely improve performance; however, investigating additional techniques to boost RCA's performance scalability may prove worthwhile in the future. Also useful would be evaluating RCA's suitability to off-chip interconnection networks, as the basic technique appears broadly applicable.

7.1.2 Low-Diameter NOC Topologies

Single-chip multiprocessors and systems-on-a-chip are highly sensitive to performance, energy, and area cost of chip-level interconnects. To date, most NOCs have featured ring- or mesh-based organizations. While these topologies map well to silicon substrates and are attractive from a complexity perspective, they incur considerable delay and energy overheads in systems with a non-trivial number of interconnected nodes. The overheads arise due to the large number of router traversals that are necessary on traffic with limited or no locality. Each router crossing incurs both latency and energy cost due buffer accesses, switch fabric traversal, and arbitration. Low diameter topologies can be used to improve connectivity; however, such networks are not always amenable to planar silicon substrates, require a large number of dedicated point-to-point channels, and may incur high router complexity.

In this thesis, we introduce a novel communication fabric developed specifically to accommodate the unique characteristics and requirements of on-chip substrates. The salient feature of the proposed Multidrop Express Channels (MECS) topology is its use of point-to-multipoint channels that provide rich inter-node connectivity with a limited number of links. Each pipelined MECS link connects the source node to multiple destinations spanned by the channel via light-weight drop interfaces. With just four multidrop channels (one per direction), a MECS router can be fully connected to other nodes in each of the X and Y dimensions. Rich connectivity and modest channel requirements improve the area and performance scalability of the MECS topology in larger network organizations.

An evaluation of MECS shows that the topology offers good area- and energy-efficiency, and is successful at minimizing communication latency at low to moderate network loads. While mesh-based networks offer superior throughput on some traffic patterns, the advantage comes at the cost of significant area and energy expense, as well as high latency at low network loads. The flattened butterfly, which is an alternative low-diameter network organization proposed for on-chip implementation, offers similar or better area and energy characteristics compared to MECS. However, the flattened butterfly requires a dedicated point-to-point channel for each pair of interconnected nodes. This attribute limits the channel width of affordable networks either through high bisection bandwidth demands or overwhelming switch complexity. As a result, the flattened butterfly shows higher serialization delays and offers lower throughput under deterministic routing than the MECS topology.

In the broader perspective, we observe that MECS belongs to a larger class of networks expressible via Generalized Express Cubes – a framework that extends k-ary n-cubes with concentration and express channels. Our analysis reveals that a number of existing NOC topology candidates can be readily expressed in this framework, which may simplify their analysis and understanding. A common framework also facilitates derivation of new topologies, which we demonstrate by evolving a new hybrid network organization with features of both MECS and flattened butterfly. We expect that additional research aimed at characterizing and classifying entire classes of networks will expand our understanding of the field and yield new insights and solutions for scalable NOC fabrics.

7.1.3 Preemptive QOS Architecture

Chip multiprocessors and systems-on-a-chip enable concurrent execution of multiple applications or virtual machines. In the process of execution, these applications and their respective threads may interfere at the NOC level in their attempts to access various on-chip resources, such as cache banks, memory controllers, and specialized accelerators. The fine-grained nature of these accesses limits of the usefulness of solutions at the software layer. Instead, single-chip systems require hardware support to provide performance isolation, fairness, and QOS guarantees.

Prior techniques for providing network quality-of-service have too much algorithmic complexity, cost (area and/or energy) or performance overhead to be attractive for on-chip implementation. In response, we propose Preemptive Virtual Clock (PVC), a light-weight QOS architecture for networks-on-chip. A distinguishing feature of PVC is its use of preemption to reduce network buffer requirements. A preemptive architecture effectively resolves in-network priority inversion situations without the need for per-flow queuing or bandwidth reservation required by previous approaches. Preemption events in a PVC network are signaled via a dedicated, low-cost acknowledgement network, enabling rapid retransmission of discarded packets by the source nodes.

An evaluation of PVC confirms that it provides fairness and supports differentiated bandwidth allocation to flows. It offers strong performance isolation, as demonstrated in a simulated denial-of-service attack against a set of parallel workloads executing on a CMP. In a 64-terminal mesh network with PVC, an average packet from the application suite experienced an 18% slowdown due to the attack traffic. In contrast, network performance under an earlier NOC QOS architecture called GSF was degraded by over 4x. PVC has a modest effect on router area by virtue of not requiring per-flow packet queues, although the scheme does require limited per-flow storage in the form of bandwidth counters for prioritization purposes. PVC's energy overhead is also unexcessive, ranging from 13 to 19% over a conventional NOC with no QOS support in a 64-terminal mesh. The greater overhead is due to preemptions which arise at high network loads.

In general, preemptions tend to diminish both performance and energy-efficiency due to the need to retransmit discarded packets, which consume network bandwidth and energy. A PVC network can employ one or more techniques to reduce preemption incidence. Packet-granularity flow control is one such mechanism that we show to be particularly effective for this purpose. Compared to flit-level flow control commonly used in existing NOCs, packet-level resource management reduces preemption incidence by a factor of three in a 64-terminal mesh network. Future research into mechanisms for reducing preemption incidence under PVC may further improve the efficiency of the scheme.

7.1.4 Kilo-node NOC Architectures

Future chips will likely integrate hundreds or even thousands of diverse components on-die. To meet the interconnect challenges of such richly integrated substrates, we examine NOC scalability with respect to area, energy, performance, and quality-of-service. Our analysis shows that low-diameter topologies in kilo-node substrates burden the network with high buffer requirements in future technology nodes. QOS overheads further increase buffer requirements and control complexity in large-scale NOCs. As a result, existing NOC architectures may fall short of meeting the efficiency demands of future chips.

To overcome the scalability limitations of existing NOCs, this thesis proposes a heterogeneous network organization. We first address an important limitation of existing QOS architectures that require hardware quality-of-service support at every router node. Instead, we introduce a topology-aware QOS architecture that isolates shared resources, such as memory controllers, in dedicated regions of the die and provides QOS support only within these regions. In doing so, we achieve a considerable reduction in router complexity in parts of the network freed from QOS overheads. The proposed scheme relies on a richly-connected topology to ensure single-hop access to QOS-enabled regions.

We also target large buffer overheads of low-diameter networks through a flow control optimization based on elastic buffering. Specifically, we develop a hybrid flow control architecture that combines elastic buffered links with virtual channel routers through a novel VC allocation mechanism. The resulting organization significantly reduces network area and energy footprint with very limited effect on performance.

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 in 15 nm technology, this design enables a 29% improvement in power and a 45% improvement in area over a state-of-the-art baseline featuring a MECS topology with virtual channel flow control and PVC QOS support throughout the network. The gains in efficiency carry a minor performance overhead at high network load and have no effect on the strength of the guarantees.

7.2 Concluding Thoughts

As process technology scaling drives greater degrees of on-chip integration, NOCs will become increasingly critical to meeting performance and efficiency objectives at the die level. To that end, this dissertation has explored techniques to improve the scalability of on-chip interconnect fabrics with respect to performance, area- and energy-efficiency, and quality-of-service. While our work advances the general state of the art in the design of richly-integrated chips from a network perspective, the proposed mechanisms have their limitations and may not always be applicable or sufficient. To that end, the rest of this chapter discusses the drawbacks of the proposed architectures and seeks to identify venues for future investigation.

7.2.1 Limitations of Proposed Techniques

Regional Congestion Awareness

In our analysis, RCA has two potential drawbacks. The first are its virtual channel requirements. Based on our evaluation, RCA does not yield any performance benefits over locally adaptive routing with two virtual channels per input port and requires up to four general-purpose VCs per port for significant performance gains. Adaptive routing based on Duato's model of deadlock avoidance [20] requires a reserved VC per input port for an escape function, which is based on dimension-order routing in our implementation. With two

VCs per port, this restriction leaves just one VC for adaptive routing, thereby complicating congestion estimation due to extreme scarcity of buffer resources. Additional VCs can greatly boost network performance under RCA as compared to DOR or locally-adaptive routing with a comparable buffer configuration; however, the gains carry an area and energy cost of extra buffers.

The other challenging aspect of RCA has to do with its applications to richly-connected topologies. While the scheme is easy to extend to support routing with a minimal number of hops, as was done in Section 4.6.3, it is not obvious how RCA can be used to prioritize paths with a non-minimal hop count. The difficulty lies in estimating the weights for various paths, where some of the paths have different numbers of intermediate routers and not all channels in a given direction lie inside the minimal routing quadrant.

Multidrop Express Channels

An obvious bottleneck of a MECS topology is the asymmetry in the number of input (many) and output (few) router ports. While this organization helps reduce router complexity and improves energy-efficiency, it limits the bandwidth through a router and hurts performance at higher load rates. The drawbacks of this design are particularly acute in systems with a large degree of concentration, which are likely to experience contention for output ports from the local interfaces, as well as under workloads that concentrate traffic through a limited number of nodes. The transpose traffic pattern discussed in Section 4.5.1 is an example of such a workload.

Several options exist for overcoming the limitations of the baseline MECS organization. One possibility, discussed in this thesis, is network replication. Multiple networks increase the channel count in a cost-effective way, but their benefits must be weighed against the increased serialization delay of narrower links. Another option is to improve the bandwidth through the switch. While a more potent switch does not relieve pressure for bandwidth out of a router, it can help performance nonetheless. In Section 4.6.2, we showed that mapping network ports to switch ports in a way that minimizes over-subscription on crossbar inputs yields a modest performance gain on uniform traffic. Increasing switch connectivity is another alternative which may be most beneficial in networks with a large degree of concentration. In such NOCs, additional switch interfaces can reduce instances of head-of-line blocking whenever traffic destined for both network and local outputs is competing for ingress switch bandwidth.

In general, the imbalance between ingress and egress bandwidth in MECS routers is integral to the topology, and is the feature that enables MECS to achieve rich connectivity and high efficiency with a modest number of links. So while it is worthwhile investigating techniques that boost performance in a MECS network, a number of other topologies will likely provide higher throughput for a given bisection wire budget.

Preemptive Virtual Clock

Energy and bandwidth losses due to preemptions are an important drawback of PVC. Since their effect on network performance and efficiency was already detailed in Sections 5.4 and 5.5, we focus on other issues here.

Implementation complexity is a potential roadblock to PVC's deployment in a real system. The distributed preemption protocol is a challenging verification target due to complex preemption scenarios involving unanticipated event sequences. For instance, a single wormhole-routed multi-flit packet may be preempted at two different routers in one cycle. This scenario carries several potential pitfalls, such as dealing with the fact that a preemption signal from the tail of the packet chasing the header flit may never find the head due to it already being discarded. To guarantee certain system properties, such as eventual delivery of every message, it may be necessary to formally verify the preemptive protocol for each design. Thus, system designers will need to weigh the benefits of PVC against the verification complexity of its distributed protocols.

Another possible limitation of PVC is its coarse latency guarantees, which are specified at the granularity of frames. Based on empirical data, we believe that frame sizes in 10's to 100's of thousands of cycles work best in systems from 64 to 1K terminals. On a chip with a 2 GHz clock, PVC would thus be able to deliver all packets within the reserved bandwidth quota in a sub-millisecond time frame for a range of NOC configurations. This result indicates that despite long frame intervals, PVC may be able to meet the latency demands of many real-time applications.

Kilo-NOC

The Kilo-NOC architecture extends MECS and PVC, and as such, it inherits their respective drawbacks. In addition, the hybrid EB/VC flow control requires its own non-trivial protocol for deadlock avoidance. The protocol has several cases that necessitate special handling, such as transfers to and from the shared regions. While likely not as complex as the preemption protocol in PVC, a practical implementation of the hybrid flow control architecture would necessitate a formal proof to guarantee deadlock freedom and extensive testing to ensure functional correctness.

7.2.2 Future Challenges and Opportunities

Future CMPs and SOCs designs will be constrained by multiple factors, including power, yield, and bandwidth limitations. To continue delivering advances in energy-efficiency, performance, and functionality that consumers have come to expect, it may be necessary to rethink various aspects of system architecture. Three items stand-out as being particularly relevant with respect to the issues covered in this thesis: (1) the need for hierarchical topologies to reduce NOC area and wiring expense; (2) use of multiple small dies on top of a silicon interposer instead of a single large die to reduce manufacturing expense and off-chip bandwidth pressure; and (3) the ability to provide chip-wide service guarantees for diverse application classes. We next discuss the challenges and opportunities associated with each of these.

Hierarchical Topologies

This thesis has focused on scaling on-chip networks to very large configurations via an essentially flat interconnect model. A flat network organization is attractive because it enhances programmability and performance stability of parallel applications, since programmers do not need to worry about bandwidth and latency implications of various levels of the interconnect hierarchy. However, a flat low-diameter network is difficult to scale due to high router complexity and significant wire demands in configurations with a large numbers of nodes.

At some point, it may make sense to abandon the flat model in favor of a hierarchical organization. A hierarchical network typically features more bandwidth and smaller communication latencies at lower levels of the interconnect hierarchy (i.e., closer to the terminals), while the opposite is true at the higher levels. By reducing network bandwidth at the global interconnect levels, hierarchical organizations enable a reduction in wire and router costs. These benefits come at a price, such as increased programming burden and load-balancing challenges for performance-sensitive parallel applications. In addition, workloads with poor locality may incur energy and delay overheads whenever accesses to global routers extend the travel path by routing away from the destination. We anticipate that hierarchical networks will become attractive once a very large number of resources is integrated on a die. Research efforts should be directed at identifying hierarchical architectures that reduce network cost and complexity while minimizing the overheads of a tiered structure.

Multi-die Packages

Very fine geometries of existing and future technologies present significant manufacturability challenges due to large variability and defect rates, which have a deleterious effect on yield and production costs. One way to overcome these challenges is to manufacture smaller dies, or slices, and “stitch” them together via a silicon interposer. The slices can be tested and binned individually, which improves yield and margins as a faulty component results in a loss of a small die instead of a large one. In addition to improving yield and reducing cost, other benefits of the approach include the ability to mix different process technologies at low cost as well as provide a high degree of product customization by combining different slices to accommodate a diverse customer base. In fact, Xilinx, a company which designs FPGA chips, has announced their intent to use this “Stacked Silicon Interposer” technology in their products [19].

System design based on slices does not necessarily require a major redesign of the interconnect. The interposer can be thought of as a dedicated high-bandwidth interconnect layer that extends the on-chip metal stack and is reachable through high-density vias crossing from the individual slices to the interposer. To reduce cost, the interposer will likely be manufactured in older technology with coarser features compared to the slices. From an interconnect perspective, coarser features are beneficial, since wires with larger dimensions have better electrical characteristics (lower resistivity and smaller RC delay), which reduces the need for repeaters or exotic signaling technologies.

One potential concern in a system with a large number of slices is the area overhead presented by vias to the interposer layer. While these vias have a finer pitch than package-level bumps, they are significantly larger than vias in the on-die metal stack. A low-diameter topology in a system with many small slices may carry a considerable area penalty due to the large number of vias required from each slice's router to the interposer. In such systems, topology with more limited connectivity, including hierarchical interconnects and high-diameter networks augmented with a limited number of express channels, may be preferred for cost reasons.

Chip-level Service Guarantees

In this thesis, we introduced interconnect-level architectures for quality-of-service. However, QOS is a system-level concern which requires understanding of QOS requirements at the application level, translating them into a set of policies at the system level, and enforcing these policies via microarchitectural mechanisms at the chip level. Significant work must be done at all three levels to provide comprehensive service guarantees. For instance, QOS requirements of virtualized web servers are likely different from those of cloud-based action games. Understanding the requirements for diverse workload classes and finding effective policies and common microarchitectural mechanisms for enforcing them will likely be a fruitful research area going forth.

Bibliography

[1] D. Abts, N. D. Enright Jerger, J. Kim, D. Gibson, and M. H. Lipasti. Achieving Predictable Performance through Better Memory Controller Placement in Many-Core CMPs. In *International Symposium on Computer Architecture*, pages 451–461, June 2009.

[2] N. R. Adiga, M. A. Blumrich, D. Chen, P. Coteus, A. Gara, M. E. Giampapa, P. Heidelberger, S. Singh, B. D. Steinmacher-Burow, T. Takken, M. Tsao, and P. Vranas. Blue Gene/L Torus Interconnection Network. *IBM Journal of Research and Development*, 49(2/3):265–276, 2005.

[3] V. Agarwal, M. S. Hrishikesh, S. W. Keckler, and D. Burger. Clock Rate Versus IPC: The End of the Road for Conventional Microarchitectures. In *International Symposium on Computer Architecture*, pages 248–259, June 2000.

[4] H. G. Badr and S. Podar. An Optimal Shortest-Path Routing Policy for Network Computers with Regular Mesh-Connected Topologies. *IEEE Transactions on Computers*, 38(10):1362–1371, 1989.

[5] 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.

[6] E. Baydal, P. Lopez, and J. Duato. A Family of Mechanisms for Congestion Control in Wormhole Networks. *IEEE Transactions on Parallel and Distributed Systems*, 16(9):772–784, 2005.

[7] L. Benini and G. De Micheli. Networks on Chips: a New SoC Paradigm. *Computer*, 35(1):70 –78, January 2002.

[8] 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*, October 2008.

[9] 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, 2006.

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

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

[12] W. J. Dally. Wire-efficient VLSI Multiprocessor Communication Networks. In *Stanford Conference on Advanced Research in VLSI*, pages 391–415, 1987.

[13] W. J. Dally and H. Aoki. Deadlock-Free Adaptive Routing in Multicomputer Networks Using Virtual Channels. *IEEE Transactions on Parallel Distributed Systems*, 4(4):466–475, 1993.

[14] 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.

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

[16] R. Das, S. Eachempati, A. Mishra, V. Narayanan, and C. Das. Design and Evaluation of a Hierarchical On-Chip Interconnect for Next-Generation CMPs. In *International Symposium on High-Performance Computer Architecture*, pages 175 –186, February 2009.

[17] 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, August 1989.

[18] T. Dieker. Simulation of Fractional Brownian Motion. Master's thesis, University of Twente, The Netherlands, 2002.

[19] P. Dorsey. Xilinx Stacked Silicon Interconnect Technology Delivers Breakthrough FPGA Capacity, Bandwidth, and Power Efficiency. `http://www.xilinx.com/support/documentation/white_papers/`, October 2010.

[20] J. Duato. A New Theory of Deadlock-Free Adaptive Routing in Wormhole Networks. *IEEE Transactions on Parallel and Distributed Systems*, 4(12): 1320–1331, 1993.

[21] J. Duato, I. Johnson, J. Flich, F. Naven, P. Garcia, and T. Nachiondo. A New Scalable and Cost-Effective Congestion Management Strategy for Lossless Multistage Interconnection Networks. In *International Symposium on High-Performance Computer Architecture*, pages 108–119, February 2005.

[22] W. Feng and K. G. Shin. Impact of Selection Functions on Routing Algorithm Performance in Multicomputer Networks. In *International Conference on Supercomputing*, pages 132–139, July 1997.

[23] D. Franco, I. Garcés, and E. Luque. A New Method to Make Communication Latency Uniform: Distributed Routing Balancing. In *International Conference on Supercomputing*, pages 210–219, May 1999.

[24] M. Galles. Scalable Pipelined Interconnect for Distributed Endpoint Routing: The SGI Spider Chip. In *HOT Interconnects IV*, pages 141–146, 1996.

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

[26] C. J. Glass and L. M. Ni. The Turn Model for Adaptive Routing. *Journal of the ACM*, 41(5):874–902, 1994.

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

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

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

[30] P. V. Gratz. *Network-On-Chip Implementation and Performance Improvement Through Workload Characterization and Congestion Awareness.* PhD thesis, The University of Texas at Austin, 2008.

[31] 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.

[32] 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.

[33] B. Grot, J. Hestness, S. W. Keckler, and O. Mutlu. Kilo-NOC: A Heterogeneous Network-on-Chip Architecture for Scalability and Service Guarantees. In *International Symposium on Computer Architecture*, pages 268–279, June 2011.

[34] K. Harteros and M. Katevenis. Fast Parallel Comparison Circuits for Scheduling. Technical Report TR-304, FORTH-ICS, March 2002.

[35] 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.

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

[37] 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.

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

[39] J. Kim, D. Park, T. Theocharides, N. Vijaykrishnan, and C. R. Das. A Low Latency Router Supporting Adaptivity for On-Chip Interconnects. In *Design Automation Conference*, pages 559–564, June 2005.

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

[41] J. H. Kim and A. A. Chien. Rotating Combined Queueing (RCQ): Bandwidth and Latency Guarantees in Low-Cost, High-Performance Networks. In *International Symposium on Computer Architecture*, pages 226–236, May 1996.

[42] K. Knauber and B. Chen. Supporting Preemption in Wormhole Networks. In *International Computer Software and Applications Conference*, pages 232–238, October 1999.

[43] 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.

[44] 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.

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

[46] A. Leon and D. Sheahan. The UltraSPARC T1: A Power-Efficient High-Throughput 32-Thread SPARC Processor. In *Asian Solid-State Circuits Conference*, pages 27 –30, November 2006.

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

[48] 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.

[49] S. S. Mukherjee, P. Bannon, S. Lang, A. Spink, and D. Webb. The Alpha 21364 Network Architecture. *IEEE Micro*, 22(1):26–35, 2002.

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

[51] 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.

[52] T. Nesson and S. L. Johnsson. ROMM Routing on Mesh and Torus Networks. In *ACM Symposium on Parallel Algorithms and Architectures*, pages 275–287, July 1995.

[53] U. Y. Ogras and R. Marculescu. Prediction-based Flow Control for Network-on-Chip Traffic. In *Design Automation Conference*, pages 839–844, July 2006.

[54] J. D. Owens, W. J. Dally, R. Ho, D. N. J. Jayasimha, S. W. Keckler, and L.-S. Peh. Research Challenges for On-Chip Interconnection Networks. *IEEE Micro*, 27(5):96–108, 2007.

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

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

[57] D. Pham, T. Aipperspach, D. Boerstler, M. Bolliger, R. Chaudhry, D. Cox, P. Harvey, P. Harvey, H. Hofstee, C. Johns, J. Kahle, A. Kameyama, J. Keaty, Y. Masubuchi, M. Pham, J. Pille, S. Posluszny, M. Riley, D. Stasiak, M. Suzuoki, O. Takahashi, J. Warnock, S. Weitzel, D. Wendel, and K. Yazawa. Overview of the Architecture, Circuit Design, and Physical Implementation of a First-Generation Cell Processor. *IEEE Journal of Solid-State Circuits*, 41(1):179–196, January 2006.

[58] RFC3393. IP Packet Delay Variation Metric for IP Performance Metrics. RFC 3393. http://www.ietf.org/rfc/rfc3393.txt.

[59] T. Ristenpart, E. Tromer, H. Shacham, and S. Savage. Hey, You, Get Off of My Cloud: Exploring Information Leakage in Third-Party Compute Clouds. In *Conference on Computer and Communications Security*, November 2009.

[60] K. Sankaralingam, R. Nagarajan, P. Gratz, R. Desikan, D. Gulati, H. Hanson, C. Kim, H. Liu, N. Ranganathan, S. Sethumadhavan, S. Sharif, P. Shivakumar, W. Yoder, R. McDonald, S. Keckler, and D. Burger. The Distributed Microarchitectural Protocols in the TRIPS Prototype Processor. In *International Symposium on Microarchitecture*, pages 480–491, December 2006.

[61] 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.

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

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

[64] S. L. Scott and G. M. Thorson. The Cray T3E Network: Adaptive Routing in a High Performance 3D Torus. In *HOT Interconnects IV*, pages 147–156, August 1996.

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

[66] 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.

[67] A. Singh, W. J. Dally, A. K. Gupta, and B. Towles. GOAL: A Load-Balanced Adaptive Routing Algorithm for Torus Networks. In *International Symposium on Computer Architecture*, pages 194–205, June 2003.

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

[69] H. Song, B. Kwon, and H. Yoon. Throttle and Preempt: A New Flow Control for Real-Time Communications in Wormhole Networks. In *International Conference on Parallel Processing*, pages 198–202, August 1997.

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

[71] D. Stiliadis and A. Varma. Design and Analysis of Frame-Based Fair Queuing: A New Traffic Scheduling Algorithm for Packet Switched Networks. In *International Conference on Measurement and Modeling of Computer Systems*, pages 104–115, May 1996.

[72] M. B. Taylor, W. Lee, S. P. Amarasinghe, and A. Agarwal. Scalar Operand Networks: On-Chip Interconnect for ILP in Partitioned Architecture. In *International Symposium on High-Performance Computer Architecture*, pages 341–353, February 2003.

[73] M. Thottethodi, A. R. Lebeck, and S. S. Mukherjee. Self-Tuned Congestion Control for Multiprocessor Networks. In *International Symposium on High-Performance Computer Architecture*, pages 107–118, January 2001.

[74] Tile-GX100. Tilera TILE-Gx100. http://www.tilera.com/products/TILE-Gx.php.

[75] B. Towles, W. J. Dally, and S. Boyd. Throughput-Centric Routing Algorithm Design. In *Symposium on Parallel Algorithms and Architectures*, pages 200–209, June 2003.

[76] L. G. Valiant. A Scheme for Fast Parallel Communication. *SIAM Journal on Computing*, 11(2):350–361, 1982.

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

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

[79] 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.

[80] 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.

[81] N. Weste and D. Harris. *CMOS VLSI Design: A Circuits and Systems Perspective*. Addison-Wesley Publishing Company, 4th edition, 2010.

[82] L. Zhang. Virtual Clock: a New Traffic Control Algorithm for Packet Switching Networks. *SIGCOMM Computer Communication Review*, 20 (4):19–29, 1990.

Vita

Boris Grot graduated with Honors from the Pennsylvania State University, where he earned the Bachelor of Science degree in Computer Science and Engineering in 2000. From 2000 to 2003, he worked as a hardware engineer in the telecommunications industry. In 2003, he enrolled at the University of California, Los Angeles and graduated in 2005 with a Master's of Science degree in Electrical Engineering. The following year, he joined the Ph.D. program in Computer Science at the University of Texas at Austin.

Permanent Address: bgrot@utexas.edu

This dissertation was typeset with LaTeX 2_ε[*] by the author.

[*] LaTeX 2_ε is an extension of LaTeX. LaTeX is a collection of macros for TeX. TeX is a trademark of the American Mathematical Society. The macros used in formatting this dissertation were written by Dinesh Das, Department of Computer Sciences, The University of Texas at Austin, and extended by Bert Kay, James A. Bednar, and Ayman El-Khashab.