Appears in the Proceedings of the 14^{th} International Symposium on High-Performance Computer Architecture

Regional Congestion Awareness for Load Balance in Networks-on-Chip

Paul Gratz[†] Boris Grot [§] Stephen W. Keckler [§]

[†] Department of Electrical and Computer Engineering
The University of Texas at Austin
pgratz@cs.utexas.edu

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

Abstract

Interconnection networks-on-chip (NOCs) are rapidly replacing other forms of interconnect in chip multiprocessors and system-on-chip designs. Existing interconnection networks use either oblivious or adaptive routing algorithms to determine the route taken by a packet to its destination. Despite somewhat higher implementation complexity, adaptive routing enjoys better fault tolerance characteristics, increases network throughput, and decreases latency compared to oblivious policies when faced with non-uniform or bursty traffic. However, adaptive routing can hurt performance by disturbing any inherent global load balance through greedy local decisions. To improve load balance in adapting routing, we propose Regional Congestion Awareness (RCA)*, a lightweight technique to improve global network balance. Instead of relying solely on local congestion information, RCA informs the routing policy of congestion in parts of the network beyond adjacent routers. Our experiments show that 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. Compared to a baseline adaptive router, RCA incurs a negligible logic and modest wiring overhead.*

1 Introduction

Moore's law has steadily increased on-chip transistor densities and enabled the integration of dozens of components on a single die. These components include regular arrays of processors and cache banks in tiled chip multiprocessors (CMPs) and heterogeneous resources in system-on-chip (SoC) designs. One outcome of greater integration is that interconnection networks have started to replace shared buses and other forms of communication featuring long, global wires. Networks-on-chip (NOCs) scale better than traditional forms of on-chip interconnect, and enjoy superior performance and fault tolerance characteristics [6]. NOCs can be constructed using nearest-neighbor point-to-point links with large bit-width, facilitating naturally-pipelined, high-bandwidth communication [14, 15].

To date, most NOCs have employed simple topologies such as two-dimensional meshes [35, 24, 34] and rings [23], in part because both designs are good matches to planar silicon manufacturing processes and support short link lengths. Minimizing router overheads, NOCs tend to use simple router designs with limited virtual channels, shallow flit buffers per virtual channel, short router pipeline stages, and messages with a limited number of flits. While the abundance of on-chip wires enables wider physical channels, wormhole flow control will likely dominate due to the shallow virtual channel buffers. NOCs tend to employ simple oblivious routing algorithms, such as dimension order routing (DOR). While such oblivious routing algorithms are easy to implement in hardware, they often do a poor job of balancing the load across the links.

Adaptive routing has been employed in multichip interconnection networks as a means to improve network performance and to tolerate network link or router failures. Despite additional implementation complexity, adaptive routing is appealing for emerging NOCs with an increasing number of connected elements. Performance can improve by routing around pockets of congestion and flattening the distribution of traffic among the links. In both cases, the improvement is realized through increased load balance, which smoothes out non-uniformities in the original traffic pattern. However, adaptive routing requires network path diversity between source 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.

The key inhibitor to performance in existing adaptive routers is an ignorance of global network state, leading to router output port selection based only on locally-available congestion estimates. Such short-sighted routing decisions tend to upset global load balance in many traffic patterns. In this paper, 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

[*] Mr. Gratz and Mr. Grot contributed equally to this paper; their names are placed alphabetically in the author list.

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. RCA has a negligible impact on router area and no impact on its critical path, compared to a conventional adaptive router.

Section 2 summarizes relevant related work in adaptive routing for both on-chip and inter-chip interconnection networks. Section 3 outlines the design of our baseline router as a point of reference and describes the new elements required for capturing congestion metrics. Section 4 describes the RCA algorithms and variants that capture different degrees of network congestion. Section 5 presents performance results of RCA along with several sensitivity studies and Section 6 concludes.

2 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 [33, 20, 32, 26]. 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 [14], leading to temporary pockets of congestion known as hotspots. Schemes that have some flexibility with respect to route choice, provide advantages over oblivious approaches that are not able to adapt to the communication pattern and network state.

Adaptive routing is a technique for fault tolerance and congestion avoidance, successfully used in commercial multiprocessors from IBM [1], Cray [25], and Compaq [18]. Non-minimal adaptive routing has the potential to improve load balance beyond the limits of minimal routing [5, 27], but at the cost of greater implementation complexity and potentially higher per-packet latency and energy. Thus, we restrict our evaluation to minimal routing, but the general principles presented here could be applied to non-minimally routed networks as well.

2.1 Routing Policies

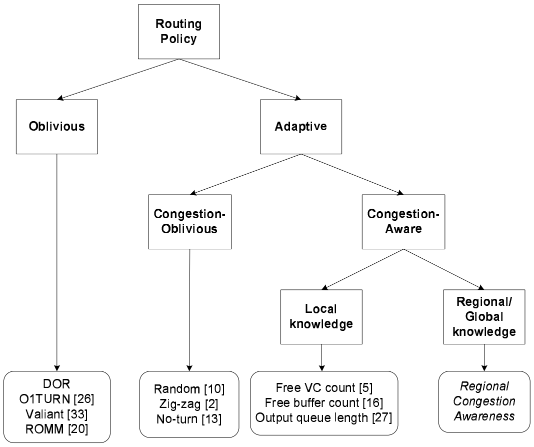


Figure 1. Taxonomy of routing policies with respect to congestion avoidance.

The routing policy determines the dynamic path taken by a given packet through an adaptively-routed network. Figure 1 presents a taxonomy of routing policies. Adaptive routing policies can be classified as either congestion-oblivious or congestion-aware, based on whether they take output link demand into account. Given a set of free and legal output ports, *random* [10] and *zigzag* [2] routing policies respectively choose an output direction randomly or based on the remaining hop count in each dimension, while *no-turn* [13] 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 [5]. 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 [16], while Singh et al. used the output queue length for the same purpose [27, 28].

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 [27] uses the queue length at each output port as its local congestion indicator during the routing phase, while GAL [28] 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.

2.2 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 [9]. 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 [11]. 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 [4, 31, 21]. 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 Network-on-Chip Routers

This section details the microarchitecture of a conventional network-on-chip router and describes the modifications necessary to support adaptivity. While we restrict the discussion to a 2D mesh, this topology is not an inherent limitation of the design.

3.1 DOR Router Microarchitecture

The canonical NOC virtual channel router was first described by Peh and Dally [22]. 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 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 [12]. Another latency-hiding approach is *speculation*, which allows switch allocation to be overlapped with VC allocation [22]. 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 reduces router latency to a single cycle if crossbar traversal is optimistically initiated in parallel with VC and switch allocation [19]. The speculation is successful only at low loads; mis-speculation incurs a one-cycle penalty. In this paper, we use a 2-cycle adaptive router design based on preselection. While we expect that the mechanisms for adaptivity are compatible with a single-cycle router, a proof is orthogonal to this work. With a one-cycle channel delay, the zero-load latency of the design is three cycles per hop for the baseline DOR router.

3.2 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 [16]. 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

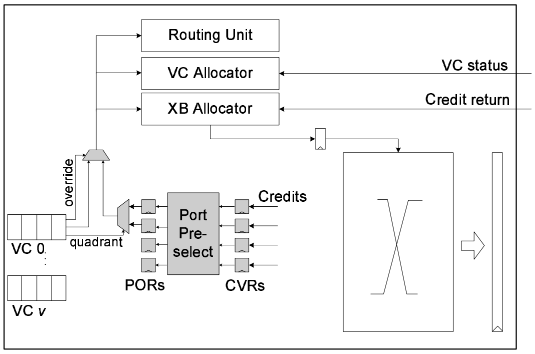


Figure 2. Two-stage adaptive router.

for each quadrant is computed every cycle for use on the following cycle based on the congestion status of each port.

Figure 2 shows the pipeline for a two-stage adaptive router based on Kim et al.'s design [16] 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 4 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.

3.3 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 [5].

Free buffers (*bf*): Kim et al. used the count of free buffers in their low-latency adaptive router [16]. 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. Since our router employs speculation, both speculative and non-speculative switch requests are counted.

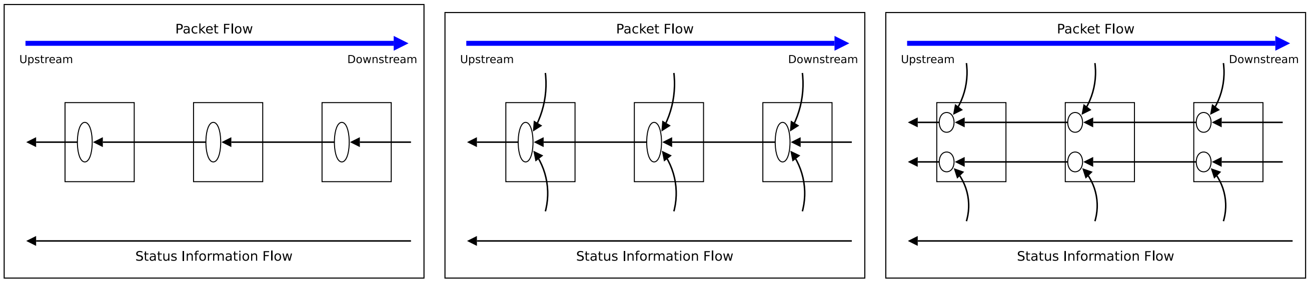
Composite metrics: Each of the atomic metrics has strengths and weaknesses. We propose simple pairings of the atomic metrics to build on their strengths and nullify their shortcomings. The three combinations of the atomic metrics are: free VCs and free buffers (*vc_bf*); free VCs and crossbar demand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

4 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 disruption this balance due to greedy, local decisions that lack knowledge of network state beyond the nearest neighbors. Adaptive routing in a 2D mesh steers traffic toward the middle of the network, leaving the edge links underutilized and congests the center of the mesh, destroying global load balance, a well-known problem shared by many existing adaptive routers.

We introduce Regional Congestion Awareness (RCA) to



(a) RCA 1D (b) RCA Fanin (c) RCA Quadrant

Figure 3. Regional Congestion Awareness

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 congestion 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.

4.1 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(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(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(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.

4.2 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, port preselection has low logic complexity, permitting integration of additional functionality with no impact on cycle time. Figure 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 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

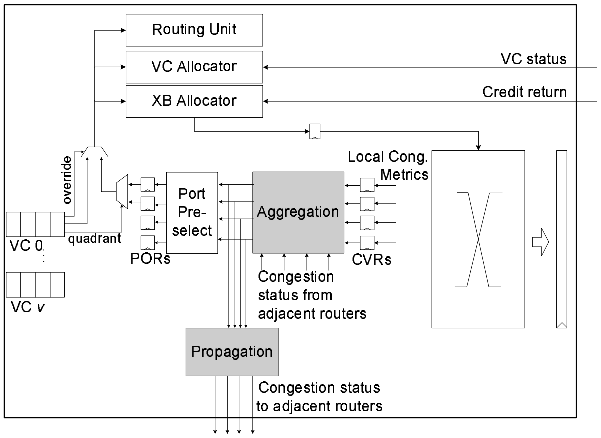
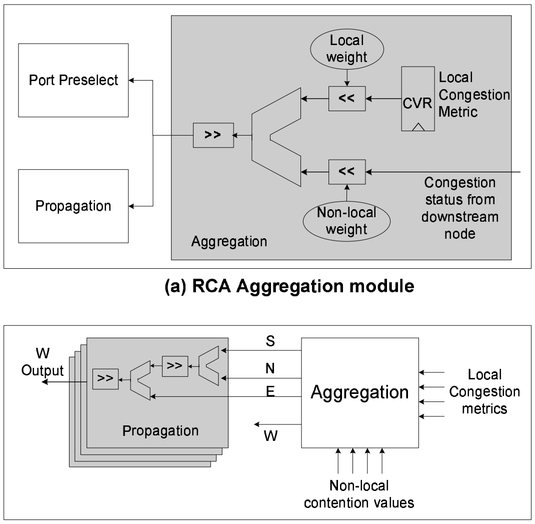


Figure 4. RCA Adaptive Router.

policy. Placing more emphasis on local congestion information moves a design toward the locally-adaptive 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.

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 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.



(a) RCA Aggregation module

(b) RCA Propagation module

Figure 5. Generic RCA aggregation module (a) and RCA Fanin propagation module (b).

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.

4.3 Status Network Design

All RCA variants must satisfy two conflicting goals: low network bandwidth and high congestion status resolution. The latter is key to early congestion detection. The averaging step in each router's aggregation unit limits the bitwidth 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 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.

Characteristic	Baseline	Variations
Topology	8x8 2D Mesh	4x4 Mesh, 16x16 Mesh
Routing	Minimal, fully-adaptive, reserved VC deadlock avoidance [8]	–
Router uArch	Two-stage speculative	–
Per-hop latency	3 cycles: 2 cycles in router, 1 cycle to cross channel	–
Virtual channels/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
Simulation warmup (cycles)	10,000	–
Analyzed packets	100,000	200,000; whole trace

Table 1. Baseline network configuration and variations.

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 [14], 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.

5 Evaluation

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

5.1 Methodology

We use a cycle-accurate network simulator that models the two-cycle router microarchitecture from Section 3. The router model is instrumented to collect the congestion metrics proposed in Section 3.3 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 1 details the baseline network configuration, along with the variations used in the sensitivity studies.

5.2 Workload

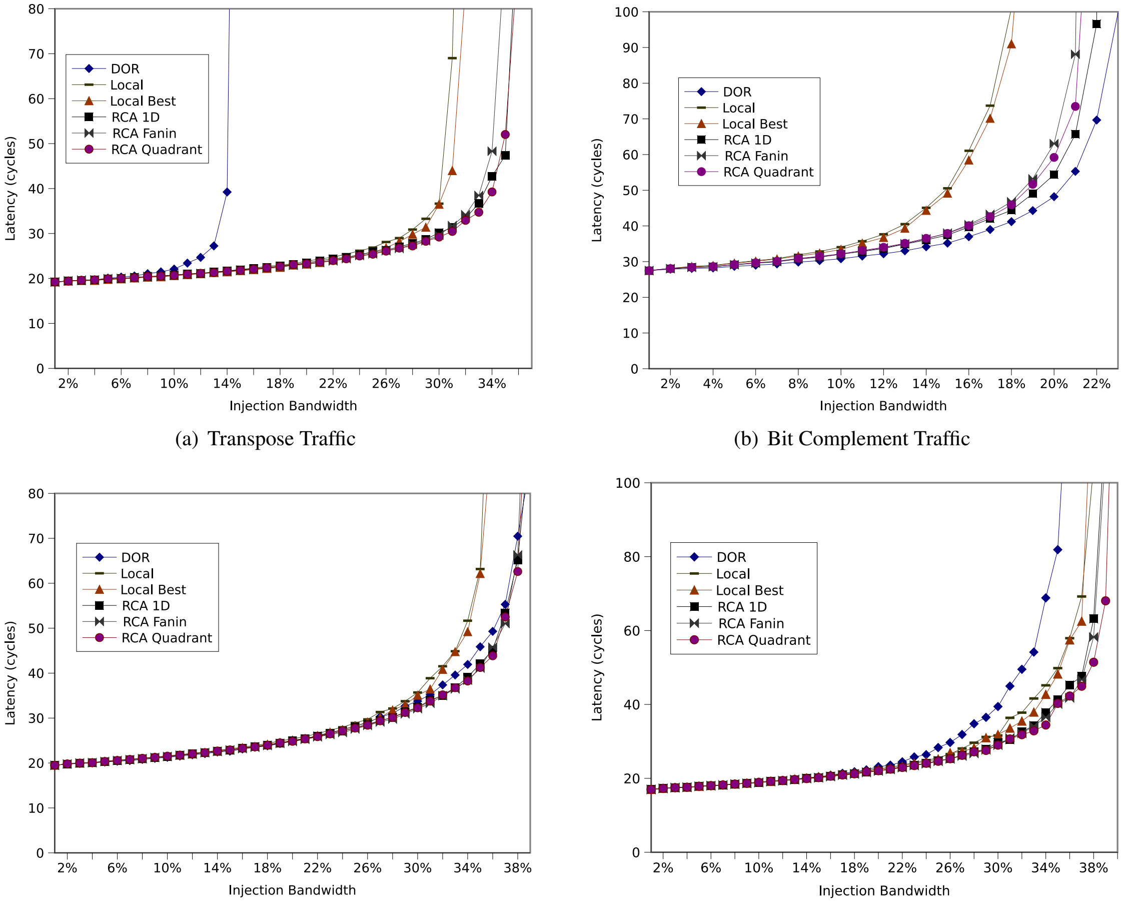
We evaluate regional congestion awareness using four standard synthetic traffic patterns: *transpose*, *bit-complement*, *uniform random* and *self-similar*. These workloads provide insight into the relative strengths and weaknesses of the different congestion metrics and aggregation techniques. They 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 [7].

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 [27].

Finally, we evaluate RCA on trace driven traffic generated from SPLASH-2 benchmarks [29], representing a typical CMP scientific workload. The traces were obtained from a forty-nine node, shared memory CMP system simulator, arranged in a 7x7 2-D mesh topology [17]. We configured our network simulator to match the environment in which the traces were captured.

5.3 Evaluation of Regional Congestion Awareness Metrics

Standard Synthetic Loads: Figure 6 contains a set of load-latency graphs for the RCA variants compared to DOR and Local across each synthetic traffic pattern. 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



(a) Transpose Traffic

(b) Bit Complement Traffic

(c) Uniform Random Traffic

(d) Self-Similar Traffic

Figure 6. Load-latency graphs comparing RCA, locally adaptive, and oblivious routing.

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 workloads, 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 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 relative

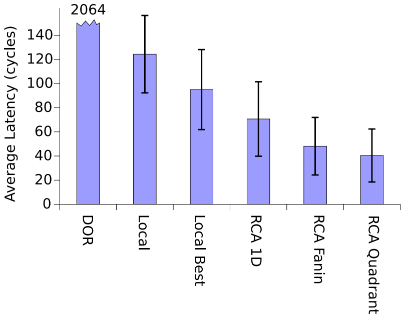


Figure 7. Average latency for 100 permutations of random pair traffic at 30% injection bandwidth. Error bars show the 95% confidence interval of the mean.

performance of the different approaches on this workload.

SPLASH-2 Benchmark Traffic: Figure 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 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.

5.4 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 show results for only the *bit-complement* traffic pattern, which we choose because, as an adversarial traffic pattern for adaptive routing, it can give us better 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 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.

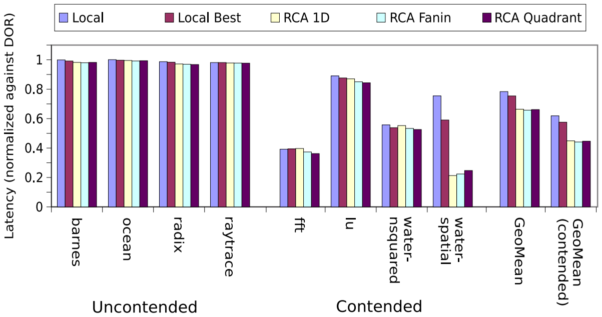


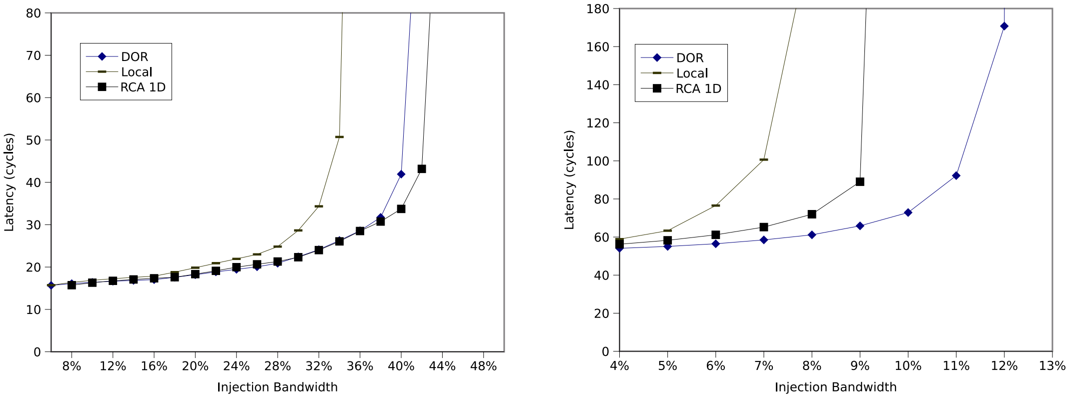
Figure 8. Average latency across SPLASH-2 benchmarks normalized to latency of DOR.

Network Dimension: On-chip networks are likely to exhibit a great deal of variation in size from design to design. Figure 9 shows load-latency graphs for two different network sizes: 4x4 and 16x16. The results for the 4x4 mesh, in Figure 9(a), show that RCA performs very well, achieving 25% better throughput than Local and 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 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, allowing RCA to maintain a lead of approximately 25%.

Packet Length: Figure 10 shows load-latency graphs for very short (1 flit) and longer (1-15 flits) packets. Short packets, shown in Figure 10(a), represent an NOC where many small values are transfered, such as in a scalar operand network [30]. Compared to the baseline *bit-complement* results in Figure 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 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.

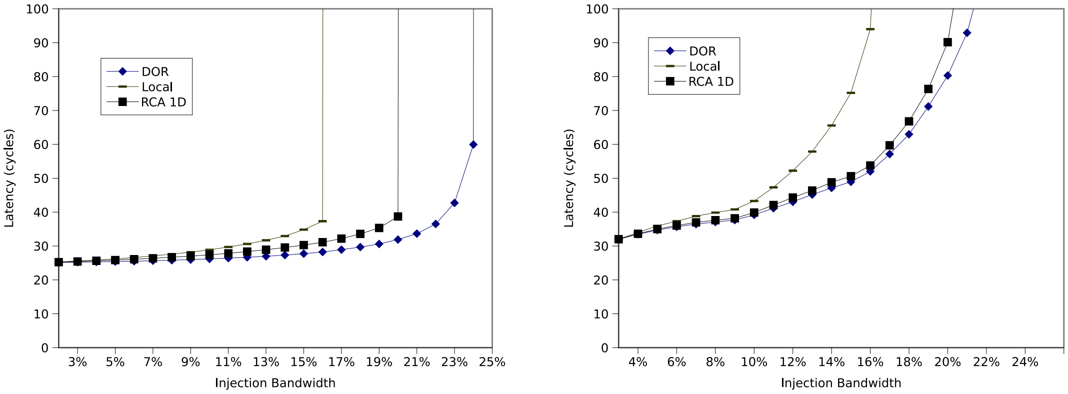
The larger distribution of packet lengths (from 1 to 15 flits) in the experiment shown in Figure 10(b) are more representative of packet sizes found in networks for memory traffic. The average packet latencies for both the adaptive and DOR routers are significantly higher for long packets than for short, 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



(a) 4x4 Mesh

(b) 16x16 Mesh

Figure 9. Load-latency graphs for 4x4 and 16x16 meshes with *bit-complement* traffic.



(a) Short Packets (1 flit)

(b) Long Packets (1-15 flits)

Figure 10. Load-latency graphs for with short and long packets with *bit-complement* traffic.

accurate picture of network utilization and improve routing decisions, almost matching the performance of DOR.

Virtual Channel Count: Figure 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.

5.5 Evaluation Summary

Across a wide range of synthetic and trace-based workloads, the RCA variants match or outperform current Local 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 [3]. Highly transient traffic patterns also complicate adaptive routing's ability to get an accurate picture of

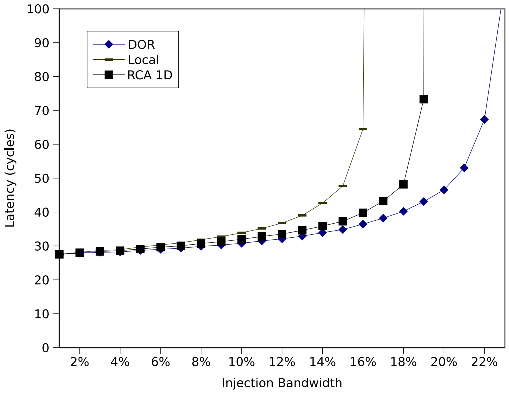


Figure 11. Load-latency graph for four virtual channels with *bit-complement* **traffic.**

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.

Although area overheads for RCA are already extremely low, as discussed in Section 4.2, we have found that RCA can *reduce* router area requirements compared to the conventional adaptive router. 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 area-constrained designs.

6 Conclusion

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 paper 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. The area overhead of RCA is minimal and its logic lies off the router's critical path. Overall, the RCA variants we examine reduce latency and improve throughput substantially over traditional local congestion metrics. Such improvements to adaptive routing can greatly enhance application performance or reduce cost at a given performance level. Furthermore, our method of aggregating and transmitting non-local congestion measurements can be applied to other minimally-adaptive routing algorithms.

While we have focused on meshes, our approach is applicable to other network topologies. For example, tori are interesting topologies as they are amenable to simple non-minimal dimension-order 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. We will examine the details of integration of RCA with non-minimal adaptive routing in our future work.

Acknowledgments

We thank Dr. Li-Shiuan Peh and Amit Kumar for their contribution of the SPLASH-2 benchmark traces. This research is supported by the Defense Advanced Research Projects Agency under contract F33615-01-C-4106 and by NSF CISE Research Infrastructure grant EIA-0303609.

References

[1] 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.

[2] 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.

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

[4] 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.

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

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

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

[8] 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.

[9] 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, 2005.

[10] 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, 1997.

[11] 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, 1999.

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

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

[14] 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, 2006.

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

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

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

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

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

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

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

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

[23] 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.

[24] 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 Microarchitecture of the TRIPS Prototype Processor. In *International Symposium on Microarchitecture*, pages 480–491, 2006.

[25] 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, 1996.

[26] 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, 2005.

[27] 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, 2003.

[28] A. Singh, W. J. Dally, B. Towles, and A. K. Gupta. Globally Adaptive Load-Balanced Routing on Tori. *IEEE Computer Architecture Letters*, 3(1):2, 2004.

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

[30] 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, 2003.

[31] 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, 2001.

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

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

[34] S. Vangal, J. Howard, G. Ruhl, S. Dighe, H. Wilson, J. Tschanz, D. Finan, P. Iyer, A. Singh, T. Jacob, S. Jain, S. Venkataraman, Y. Hoskote, and N. Borkar. An 80-Tile 1.28 TFLOPS Network-on-Chip in 65nm CMOS. In *IEEE International Solid-State Circuits Conference*, pages 98–99, February 2007.

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