

Holistic Routing Algorithm Design to Support Workload Consolidation in NoCs

Sheng Ma, *Student Member, IEEE*, Natalie Enright Jerger, *Member, IEEE*,
Zhiying Wang, *Member, IEEE*, Mingche Lai, *Member, IEEE*, and Libo Huang

Abstract—To provide efficient, high-performance routing algorithms, a holistic approach should be taken. The key aspects of routing algorithm design include adaptivity, path selection strategy, VC allocation, isolation and hardware implementation cost; these design aspects are not independent. The key contribution of this work lies in the design of a novel selection strategy, Destination-Based Selection Strategy (DBSS) which targets interference that can arise in many-core systems running consolidation workloads. In the process of this design, we holistically consider all aspects to ensure an efficient design. Existing routing algorithms largely overlook issues associated with workload consolidation. Locally adaptive algorithms do not consider enough status information to avoid network congestion. Globally adaptive routing algorithms attack this issue by utilizing network status beyond neighboring nodes. However, they may suffer from interference, coupling the behavior of otherwise independent applications. To address these issues, DBSS leverages both local and non-local network status to provide more effective adaptivity. More importantly, by integrating the destination into the selection procedure, DBSS mitigates interference and offers dynamic isolation among applications. Results show that DBSS offers better performance than the best baseline selection strategy and improves the energy-delay product for medium and high injection rates; it is well suited for workload consolidation.

Index Terms—Networks-on-chip, Adaptive Routing Algorithm, Workload Consolidation.



1 INTRODUCTION

Given the difficulty of extracting parallelism, it is quite likely that multiple applications will run concurrently on a many-core system [3], [25], [35], often referred to as workload consolidation. Significant research exists on maintaining isolation and effectively sharing on-chip resources such as caches [50] and memory controllers [43]. The network-on-chip (NoC) [12] is another, less-explored example of a shared resource where one application's traffic may degrade the performance of another. This work focuses on improving performance and providing isolation for workload consolidation via the routing algorithm.

There are several requirements placed on the routing algorithm for high performance. First, the routing algorithm should leverage available path diversity to provide sufficient adaptivity to avoid network congestion. Closely related to the issue of adaptivity is virtual channel (VC) [9] allocation and the need to provide deadlock freedom. Second, it should not leverage superfluous information leading to inaccurate estimates of network status. Finally, the routing algorithm should be implemented with low hard-

ware cost. Workload consolidation has an additional requirement: dynamic isolation among applications. Existing routing algorithms are unable to meet all these needs. Oblivious routing, such as DOR, ignores network status, resulting in poor load balancing. Adaptive routing offers the ability to avoid congestion by supporting multiple paths between a source and destination; a selection strategy is applied to choose between multiple outputs. Most existing selection strategies do not offer both adaptivity and isolation.

The selection strategy should choose the output port that will route a packet along the least congested path. A local selection strategy leverages only the status of neighboring nodes, which tends to violate the global balance intrinsic to traffic [22]. Globally adaptive selection strategies, such as Regional Congestion Awareness (RCA) [22], utilize a dedicated network to gather global information; it introduces excess information when selecting the output port and offers no isolation among applications, leading to performance degradation for consolidated workloads. This excess information can be classified as intra- and inter-application interference. Interference makes the performance less predictable.

Considering the future prevalence of server consolidation and the need for performance isolation, an efficient routing algorithm should combine high adaptivity with dynamic workload isolation. Therefore, we believe utilizing precise information is preferable; redundant or insufficient information easily leads to inferior performance. Based on this insight, we introduce Destination-Based Selection Strategy (DBSS),

- Sheng Ma, Zhiying Wang, Mingche Lai and Libo Huang are with the School of Computer, National University of Defense Technology, Changsha, Hunan, China.

This research was carried out while Sheng Ma was a visiting international student at the UofT supported by a CSC scholarship.

E-mail: {masheng, zywang, mingchelai, libohuang}@nudt.edu.cn

- Natalie Enright Jerger is with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, Ontario, Canada.
E-mail: enright@eecg.toronto.edu

which is well suited to workload consolidation.

We design a low-cost congestion information propagation network to leverage both local and non-local network status, giving DBSS high adaptivity. Furthermore, DBSS chooses the output port by only considering the nodes that a packet may traverse, while ignoring nodes located outside the minimum quadrant defined by the current location and the destination. Thus, it eliminates redundant information and dynamically isolates applications.

Integral to the evaluation of a new selection strategy is the underlying routing function. To provide a thorough exploration, we analyze the offered path diversity of several fully and partially adaptive routing functions. We also consider the VC re-allocation scheme [13] that is required for deadlock freedom. Based on an appropriate routing function, we evaluate DBSS against other selection strategies for both a regular and concentrated mesh. Our experimental results show that DBSS outperforms other strategies for all evaluated network configurations.

This paper makes the following contributions:

- Analyzes the limitations of other selection strategies and proposes DBSS which affords sufficient adaptivity for congestion avoidance and dynamic isolation among applications.
- Explores the effect of interference and demonstrates that the amount of congestion information considered impacts performance, especially for consolidated workloads.
- Designs a low-cost congestion status propagation network with only 3.125% wiring overhead to leverage both local and non-local information.
- Holistically considers path diversity and VC re-allocation to provide further insight for routing algorithm design.

2 BACKGROUND

In this section, we discuss related work in application mapping and adaptive routing algorithms design.

Since the arrival order and execution time of consolidated workloads cannot be known at design time, run-time application mapping techniques are needed [7], [8], [27], [33]. Mapping each application to a near convex region is optimal for workload consolidation [7], [8]. Most application mapping techniques consider the Manhattan distance between the source and destination but not the routing paths [8], [33]; routing algorithm design is complementary to these techniques.

As shown in Fig. 1, an adaptive routing algorithm consists of two parts: the routing function and the selection strategy [1]. The routing function computes the set of possible output channels, while the selection strategy chooses one of them based on the network status. A routing function must achieve deadlock freedom. This can be achieved by removing cycles from

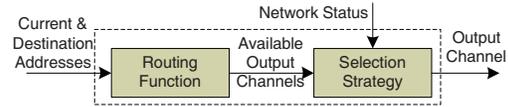


Fig. 1. The structure of an adaptive routing algorithm.

the channel dependency graph [11]. Duato proved that cycles are allowed in the channel dependency graph once there is a deadlock-free routing sub-function [15], [16]. Duato's theory is powerful for designing fully adaptive routing functions which allow all minimal paths for packet routing. Turn model functions avoid deadlock by offering only a subset of all minimal paths [6], [19], [21]. In addition to the offered path diversity, partially and fully adaptive functions differ in the VC re-allocation. Partially adaptive routing can utilize an aggressive VC re-allocation scheme, while fully adaptive routing can only leverage a conservative scheme. VC re-allocation strongly impacts performance and cannot be overlooked in the design of routing algorithms [19], [37].

Off-chip networks are constrained by pin bandwidth, but the abundant wiring resources in NoCs allow easier implementation of congestion propagation mechanisms. Therefore, the NoC paradigm has sparked renewed interest in adaptive routing algorithms. DyAD combines the advantages of both deterministic and adaptive routing schemes [26]. DyXY uses dedicated wires to investigate the status of neighboring routers [34]. A low-latency adaptive routing algorithm performs lookahead routing and pre-selects the optimal output port [30]. The selection strategies of these designs [26], [30], [34] all leverage the status of the neighboring nodes. Many oblivious selection strategies have also been evaluated including zigzag, XY, and no turn [10], [18], [21], [39], [49]. Neighbors-on-Path (NoP) makes a selection based on the condition of the nodes adjacent to neighbors [1].

RCA is the first work utilizing global information to improve load balancing in NoCs [22]. However, it introduces interference. Redundant information may degrade the quality of the congestion estimates; to combat this, packet destination information can be leveraged to eliminate excess information [46]. Per-destination delay estimates steer the output selection. Despite leveraging a similar observation, our design is quite different. Moreover, we focus on the performance with workload consolidation. BLBDR [47] provides strict isolation among applications by statically configuring connectivity bits. In contrast, we offer more dynamic isolation. Part of this research was presented at ISCA 2011 [36].

3 MOTIVATION

We motivate the need for a novel selection strategy from two directions. First, the selection strategy should have enough information about network conditions to offer effective congestion avoidance. Both

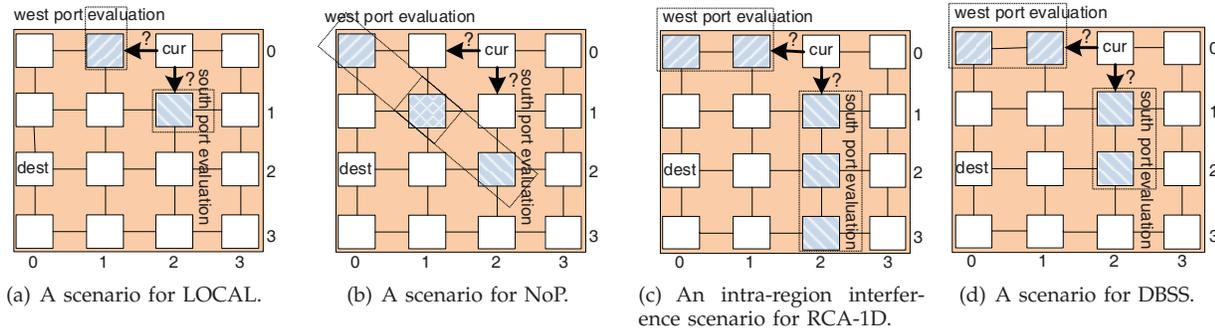


Fig. 2. Packet routing example. The current router is (0,2) and the destination is (2,0).

local and NoP [1] selection strategies lack enough information, leading to sub-optimal performance. Second, intra- and inter-application interference should be minimized. RCA [22] utilizes global network information but does not consider interference. DBSS offers a middle ground between these extremes.

3.1 Insufficient information

The local selection strategy (LOCAL) leverages the conditions of neighboring nodes. These conditions may be free buffers [1], [22], [26], [30], [34], free VCs [10], [22], crossbar demands [22] or a combination [22]. Fig. 2(a) shows a packet at router (0,2) that is routed to (2,0). A selection strategy is needed to choose between the west and south ports; LOCAL only uses the information about the nearest nodes ((0,1) and (1,2)). Without any information about the condition beyond neighbors, it cannot avoid congestion more than one hop away from current node.

The NoP uses the status of nodes adjacent to neighboring nodes as shown in Fig. 2(b). The limitation is that NoP ignores the status of neighbors ((0,1) and (1,2)); it makes decisions based only on the conditions of nodes two hops away. In the example, for west output evaluation, it considers nodes (0,0) and (1,1). For south port, it considers nodes (2,2) and (1,1). This strategy works well with an odd-even turn model [1]. However, with a fully adaptive routing function, its performance degrades due to limited knowledge.

3.2 Intra-region interference

Three RCA variants have been proposed: RCA-1D, -Fanin and -Quadrant [22]. No single RCA variant provides the best performance across all traffic patterns; therefore, we use RCA-1D as a baseline. Fig. 2(c) shows an intra-region¹ scenario for RCA-1D. All 16 nodes run the same application. When evaluating the west output, RCA-1D considers nodes (0,1) and (0,0). Similarly, it considers nodes (1,2), (2,2) and (3,2) when evaluating the south port. For destination node (2,0), the information from node (3,2) causes interference as it lies outside the minimum quadrant defined by (0,2) and (2,0); the packet will not traverse this node. This

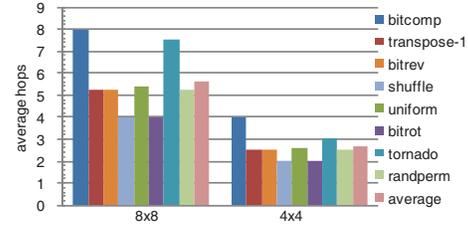


Fig. 3. The average hop count for synthetic traffic.

interference may result in poor output port selection, especially considering traffic locality.

We show the average hop count (AHP) to measure locality for several synthetic traffic patterns [13] in Fig. 3. Most traffic has an AHP of less than 5.6 (average 5.58) and 3 (average 2.63) for 8×8 and 4×4 meshes respectively. These patterns exhibit locality as most packets travel a short distance. Thus, we need strategies to mitigate intra-application interference.

3.3 Inter-region interference

Fig. 4 illustrates a workload consolidation example for an 8×8 mesh; similar scenarios will be prevalent in many-core systems. Here, there are 4 concurrent applications and each one is mapped to a 4×4 region. Region R0 is defined by nodes (0,0) and (3,3), R1 is defined by (0,4) and (3,7), R2 is defined by (4,0) and (7,3), and R3 is defined by (4,4) and (7,7). Fig. 4 shows a packet in R0 whose current location and destination are (0,2) and (2,0), respectively. Even though traffic in R0 is isolated from traffic in other regions, RCA-1D considers the congestion status of nodes in R2 when selecting output ports for traffic belonging to R0. Obviously, this method introduces interference and reduces performance isolation.

To evaluate the effect of inter-region interference, we assign transpose-1 traffic [6], [13] to R0 and uniform random traffic to R1-R3². Fig. 5 shows the performance of R0. The 'RCA-uni_region' curve represents a single region (R0) in a 4×4 network; this latency reflects perfect isolation and no inter-region interference. The saturation throughput of RCA-1D is $\sim 65\%$. However, without isolation, the saturation throughput drops dramatically to $\sim 50\%$, as shown with the 'RCA-multi_regions(4%)' curve where R1-R3 all have a 4%

1. We use region and application interchangeably.

2. See Sec. 6 for full experimental methodology.

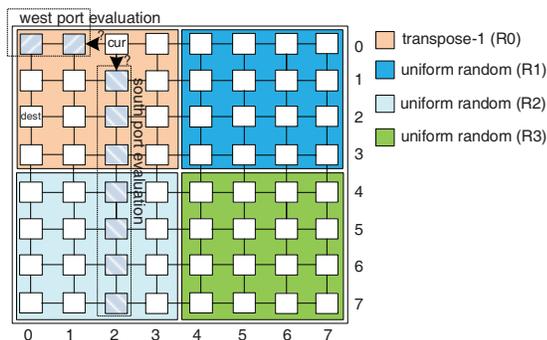


Fig. 4. An inter-region interference scenario for RCA-1D. The current router is (0,2) and destination is (2,0).

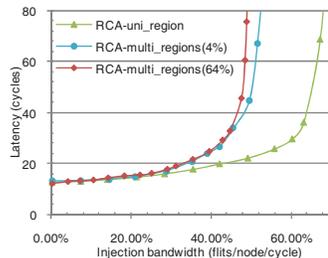


Fig. 5. Load-latency graph of region 0.

injection rate. For the ‘RCA-multi_regions(64%)’ curve, R1 has a 64% injection rate while R2 and R3 remain at 4%; in this unbalanced scenario, the saturation throughput of R0 further decreases to 47%. Clearly, the congestion information of R1-R3 greatly affects the routing selection in R0. RCA-1D couples the activities of otherwise independent applications, which is not desirable for workload consolidation.

DBSS aims to reduce the interference by considering only nodes in the minimum quadrant defined by the current and destination nodes. In Fig. 2(d), when evaluating the west output, DBSS considers nodes (0,1) and (0,0); for the south output, it considers (1,2) and (2,2). DBSS leverages information from both local and non-local nodes; it has more accurate knowledge than LOCAL and NoP. DBSS does not consider node (3,2) which eliminates interference. More importantly, for workload consolidation with each application mapped to a near convex region [7], [8], DBSS dynamically isolates routing for each region.

4 DBSS SELECTION STRATEGY

The selection strategy in adaptive routing algorithms significantly impacts performance [1], [18], [39], [49]. An efficient strategy should ideally satisfy two goals: high adaptivity and dynamic isolation.

4.1 Congestion Information Propagation Network

NoCs can take advantage of abundant wiring to employ a dedicated network to exchange congestion status. First we explain the design of the low-cost congestion information propagation network, which enables a router to leverage both local and non-local status. Both NoP and RCA utilize such a low-bandwidth monitoring network [1], [22]. We focus

on obtaining global information; the propagation network in RCA serves as the best comparison point.

At each hop in RCA’s congestion propagation network, the local status is aggregated with information from remote nodes and then propagated to upstream routers [22]. This implementation has two limitations. First, the aggregation logic combines local and distant information during transmission, making it impossible to filter out superfluous information. Second, the aggregation logic adds an extra cycle of latency per hop, leading to stale information at distant routers. Based on these two observations, we propose a novel propagation network, which consumes only one cycle per tile, giving DBSS timelier network status. More importantly, the design makes it feasible to filter out information based on the packet destination.

Fig. 6 shows the congestion propagation network for the third row of an 8×8 mesh; the same structure is present in each row and column. Along a dimension, each router has a register (*congestion_X* or *congestion_Y*) to store the incoming congestion information. The incoming information along with the local status are forwarded to the neighboring nodes in the next cycle via congestion propagation channels.

We use free VC count as the congestion metric. To cover the range of free VCs, the width of each congestion propagation channel along one direction is $\log(\text{numVCs})$. However, a coarser approximation is sufficient. For neighboring routers, making a fine distinction has little impact. On the other hand, since the incoming congestion information is weighed according to the distance, it is also unnecessary to have accurate numbers for distant routers. As we show in Sec. 9, one wire per node is sufficient to achieve high performance; the router forwards congestion information in an on/off manner. The threshold for indicating congestion is 4 (out of 8 VCs); when 5 or more VCs are available no congestion is signaled. Coarse-grain congestion signals will toggle infrequently resulting in a low activity factor and low power consumption.

Using this coarse-grain monitoring, both the *congestion_X* and *congestion_Y* registers are $n + 1$ bits wide for an $n \times n$ network (e.g. 9 bits for an 8×8 network). Incoming congestion information from routers in the same dimension is stored in 7 bits and the other 2 bits store the conditions of two ports of the local router. The router weighs the incoming congestion information based on the distance from current router; the weight is halved for each additional hop. This ratio is chosen based on prior work [22] and implementation complexity. Adjacent bit positions of a register inherently maintain a step ratio of 0.5; we implement this weighting by putting the incoming information into the appropriate positions in the register.

Fig. 7 shows the format of *congestion_X* in router (3,2). Bit 0 stores the east input port status of current router. Bits 1 and 2 store the incoming congestion information from its nearest and one hop farther west

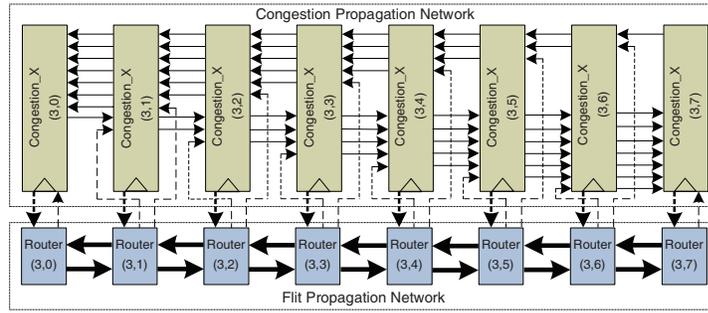


Fig. 6. The congestion information propagation network along one dimension. Bold arrows represent channels of multiple bits. Thin arrows represent channels of one bit.

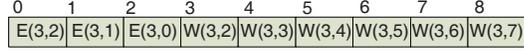


Fig. 7. Format of *congestion_X* (E: East, W: West).

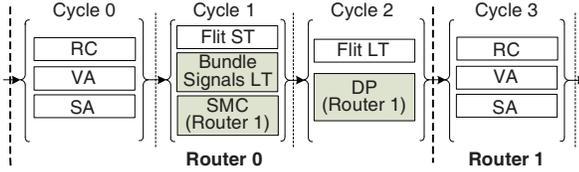


Fig. 8. The pipeline of DBSS router.

neighbor: (3,1) and (3,0), respectively. These three bits are forwarded to the east neighbor: (3,3). Bit 3 stores the west input port status of current router, and the following five bits sequentially store the west input port status of the remaining east side routers based on distance. These six bits are forwarded to the west neighbor: (3,1). *congestion_Y* has a similar format.

4.2 DBSS Router Microarchitecture

DBSS router is based on a canonical VC router [13], [17]. The router pipeline has four stages: routing computation (RC), VC allocation (VA), switch allocation (SA) and switch traversal (ST). Link traversal (LT) requires one cycle to forward the flit to next hop. For high performance, the router uses speculative switch allocation [45]; VA and SA proceed in parallel at low load. We also leverage look-ahead adaptive routing computation; the router calculates at most two alternative output ports for the next hop [20], [22], [30]. Advanced bundles [24], [31] encoding the packet destination traverse the link to the next hop while the flit is in the ST stage as shown in Fig. 8.

Selection Metric Computation. The Selection Metric Computation (SMC) and Dimension Pre-selection (DP) modules are added as illustrated in Fig. 9. SMC computes the dimension of the optimal output port for every possible destination using the congestion information stored in *congestion_X* and *congestion_Y*. An additional register, *out_dim* stores the results of SMC. With minimal routing, there are at most two admissible ports (1 per dimension) for each destination. Due to this property, the *out_dim* register uses one bit to represent the optimal output port. If the value is '0', the optimal output port is along dimension X; otherwise, it is along dimension Y.

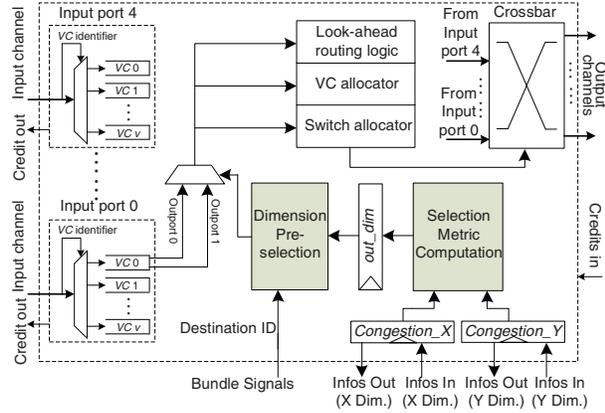


Fig. 9. DBSS router architecture.

Fig. 10 shows the pseudo-code of SMC to compute the optimal output dimension for a packet whose destination is the pos^{th} bit position of *out_dim* register. Packets forwarded to the local node are excluded from this logic. Along each dimension, only those bit positions in *congestion_X* and *congestion_Y* storing congestion information for nodes inside the quadrant defined by the current and the pos^{th} nodes are chosen. The chosen values are the congestion status metric for each dimension. According to the relative magnitude of the congestion status for the X and Y dimensions, SMC sets the value of the pos^{th} bit in *out_dim* register. If their magnitudes are equal, DBSS randomly chooses an output dimension.

Dimension Pre-selection. To remove the output port selection from the critical path, the DP module accesses the *out_dim* register one cycle ahead of the flit's arrival. The value of *out_dim* is computed out by SMC in the previous cycle. The DP module is implemented as a 64-to-1 multiplexer, which selects the corresponding bit position of *out_dim* register according to the destination encoded in an advanced bundle. When the head flit arrives, it chooses the output port according to the result of DP. Using the logical effort model [45], the delay of the DP module is ~ 8.1 FO4. If the DP were added to the VA stage, the critical path would increase from 20 FO4 to 28.1 FO4. Advanced bundles serve to avoid this increase.

```

1: if ( pos_x < cur_x )
2:   tmp_x[0:cur_x-pos_x-1] ← congestion_X[1:cur_x-pos_x];
3: else if ( pos_x > cur_x )
4:   tmp_x[0:pos_x-cur_x-1] ← congestion_X[cur_x+2:pos_x+1];
5: else { out_dim[pos] ← 1; return;}
6: if ( pos_y < cur_y )
7:   tmp_y[0:cur_y-pos_y-1] ← congestion_Y[1:cur_y-pos_y];
8: else if ( pos_y > cur_y )
9:   tmp_y[0:pos_y-cur_y-1] ← congestion_Y[cur_y+2:pos_y+1];
10: else { out_dim[pos] ← 0; return;}
11: if( tmp_x < tmp_y ) out_dim[pos] ← 1;
12: else if( tmp_x > tmp_y ) out_dim[pos] ← 0;

```

Fig. 10. The pseudo-code of SMC. (cur_y , cur_x) and (pos_y , pos_x) are the positions of current and pos^{th} router respectively. The initial value of tmp_x and tmp_y are 7-bit 0s.

5 ROUTING FUNCTION DESIGN

The focus of this work is a novel selection strategy; however, the efficacy of the selection strategy is tightly coupled to other aspects of routing algorithm design included adaptivity and VC re-allocation. We compare several adaptive routing functions. Previous analysis [41] ignores the effect of VC re-allocation. The majority of NoC packets are short [23], [37], making VC re-allocation more important. Thus, it is necessary to re-evaluate routing functions by considering both the offered path diversity and VC re-allocation.

5.1 Offered path diversity

Partially adaptive routing functions avoid deadlock by forbidding certain turns [6], [19], [21]. Due to the prohibition of east→south and north→west turns, negative-first routing can utilize all minimal paths when the X and Y positions of the destination are both positive or negative to the source. If only one is positive, it provides one minimal path [21]. Similarly, west-first and north-last utilize all minimal paths when the destination is east and south of the source, respectively. Otherwise, only one path is allowed [21]. The odd-even applies different turn restrictions on odd and even columns for even adaptiveness.

In contrast to turn models, fully adaptive routing provides all minimal physical paths but places restrictions on VCs. Most fully adaptive routing functions are designed based on Duato's theory [15], in which the VCs are classified into adaptive and escape VCs. There is no restriction on the routing of adaptive VCs; escape VCs can be only utilized if the output port adheres to a deadlock-free algorithm, usually DOR.

We compare the path diversity offered by a fully adaptive routing function (Duato) with several turn models in Fig. 11. The path diversity is measured as the ratio of times that the routing function provides two admissible ports versus one. We vary the VC count. For turn models, one VC is enough to avoid deadlock; Duato needs at least two VCs. A local selection strategy is utilized. When each port is configured with four or fewer VCs, the selection is based on the buffer availability of neighbors. The VC availability is utilized if the VC count is greater than four. Adjusting the congestion metric according to the

VC count can more stringently reflect the network status. The experiments are conducted on a 4×4 mesh; larger networks show similar trends.

As can be seen, at least $\sim 25\%$ of routing computations produce two admissible ports; a selection strategy is needed to make a choice. For Duato and negative-first, more than 50% of routing computations utilize the selection strategy under bit reverse and transpose-1 patterns. This ratio increases with larger network size. These results emphasize the importance of designing an effective selection strategy.

Duato generally shows the highest path diversity. The only exception is for transpose-1. Since all traffic is between the north-east and south-west quadrants, negative-first has the highest path diversity. The limitation on escape VCs yields slight lower diversity for Duato. Once a packet enters an escape VC, it can only use escape VCs in subsequent hops; this packet loses path diversity. For the same reason, the diversity of Duato decreases with smaller VC counts. There is almost no difference for Duato with six and eight VCs; Six VCs are enough to mitigate the escape VC limitation. The path diversity of partially adaptive routing is not sensitive to VC count. Negative-first offers no path diversity for transpose-2 since all traffic is between the north-west and south-east quadrants. Negative-first offers unstable path diversity for two symmetric transpose patterns. The four evaluated patterns are symmetric around the center of the mesh; west-first and north-last perform the same under such patterns.

5.2 VC Re-allocation Scheme

The VC re-allocation is an important yet often overlooked limitation for fully adaptive routing functions. Due to cyclic channel dependencies, only empty VCs can be re-allocated [15], [16]. If non-empty VCs are re-allocated, a deadlock configuration is easily formed [15], [16].

Since partially adaptive functions prohibit cyclic channel dependencies, non-empty VCs can be re-allocated [11]. They re-allocate an output VC when the tail flit of last packet goes through the ST stage of *current* router, which is called aggressive VC re-allocation. Instead, fully adaptive functions re-allocate an output VC only after the tail flit of last packet goes through the ST stage of *next-hop* router, which is called conservative VC re-allocation [13], [19]. The difference between these schemes is shown in Fig. 12.

In Cycle 0, Packet P_0 resides in VC_0 . P_0 waits for VC_2 and VC_3 , which are occupied by P_1 and P_2 , respectively. Both VC_2 and VC_3 have some free slots. The aggressive VC re-allocation scheme forwards the header flit of P_0 to VC_2 in Cycle 1, as shown in Fig. 12(b). However, for fully adaptive routing, P_0 must wait for $3+i$ cycles to be forwarded (Fig. 12(c)), where i denotes the delay due to contention. Assuming round-robin arbitration [13] for both VA and SA

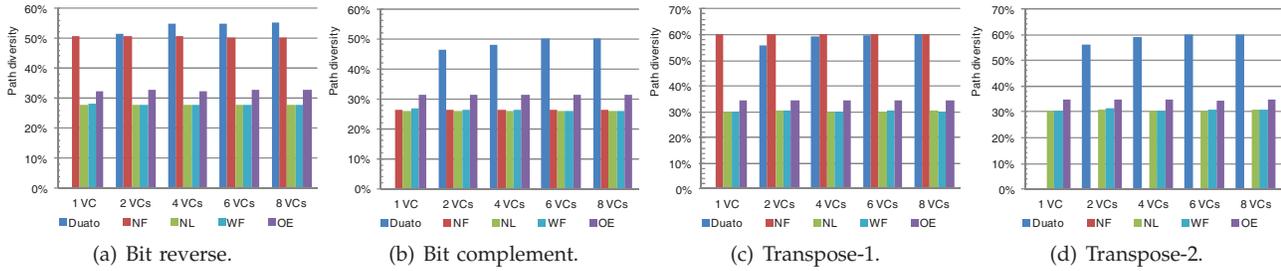


Fig. 11. Path diversity offered by several routing functions. (Duato: a fully adaptive function based on Duato’s theory; NF: negative-first; NL: north-last; WF: west-first; OE: odd-even. Duato is not shown with 1 VC as it requires at least 2 VCs for deadlock freedom. NF shows zero path diversity for transpose-2.)

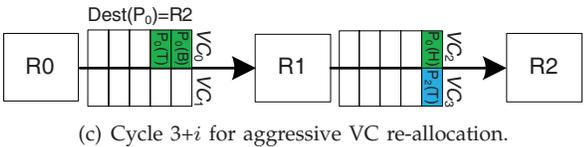
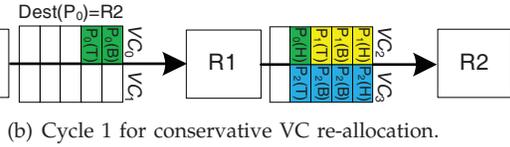
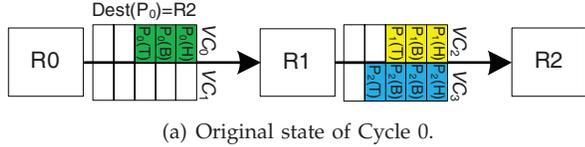


Fig. 12. The difference between two VC re-allocation schemes. ($P_i(H)$, $P_i(B)$, $P_i(T)$): the header, body and tail flit of P_i , respectively.)

and no contention from other input ports, it takes 6 cycles for VC_2 to become empty. Then, P_0 can be forwarded. More cycles are needed with contention from other input ports. The aggressive VC re-allocation has better VC utilization leading to improved performance. Allowing multiple packets to reside in one VC may result in Head-of-line (HoL) blocking [28]. However, as we will show in Sec. 6.1, with limited VCs, making efficient use of VCs strongly outweighs the negative effect of HoL blocking.

6 EVALUATION

We modify BookSim [13] to model the microarchitecture discussed in Sec. 4.2. The evaluation consists of three parts: we first consider the routing functions. Negative-first, north-last, west-first, odd-even models and a fully adaptive routing function based on Duato’s theory are evaluated with a local selection strategy. Then, the fully adaptive function is chosen for the selection strategy evaluation. A local strategy (LOCAL), NoP, RCA-1D³ and DBSS are implemented for different size meshes. Finally, we extend the evaluation to concentrated meshes [2], [32].

Several synthetic traffic patterns [6], [13] are used. Each VC is configured with 5 flit buffers, and the

TABLE 1. Full system simulation configuration.

# of cores	16
L1 cache (D & I)	private, 4-way, 32KB each
L2 cache	private, 8-way, 512KB each
Cache coherence	MOESI distributed directory
Topology	4×4 Mesh, 4 VNs, 8 VCs/VN

packet length is uniformly distributed between 1 and 6 flits. Workload consolidation scenarios are evaluated with multiple regions configured inside a network. The injection procedure of each region is independently controlled as if a region is a whole network. The destinations of all traffic generated from one region stay within the same region. The latency and throughput is measured for each region. The routing algorithm is configured at the network level.

To measure full-system performance, we leverage FeS2 [44] for x86 simulation and BookSim for NoC simulation. FeS2 is implemented as a module for Virtutech Simics [38]. We run PARSEC benchmarks [4] with 16 threads on a 16-core CMP, organized in a 4×4 mesh. Workload consolidation scenarios are also evaluated. An 8×8 mesh with four 4×4 regions is configured in BookSim. Region R0 delivers the traffic generated by FeS2, while the remaining regions run uniform random patterns. A mix of real applications and synthetic traffic was used due to scalability problems with the simulator and operating system [5]. Prior research shows the frequency of simple cores can be optimized to 5~10 GHz, while the frequency of NoC routers is limited by the allocator speed [14]. We assume cores are clocked 4× faster than the network. Cache lines are 64 bytes, and the network flit width is 16 bytes. All benchmarks use the *sims*small input sets. The total runtime is used as the performance metric. Table 1 presents the system configuration.

6.1 Evaluation of Routing Functions

Fig. 13 illustrates the results of routing function evaluation; the achieved saturation throughput⁴ is the performance metric. Comparing partially adaptive functions in Figs. 11 and 13, generally higher path diversity leads to higher saturation throughput. However, due to the conservative VC re-allocation, Duato’s

3. RCA-1D is referred to as RCA throughout the evaluation.

4. The saturation point is when the average latency is 3 times the zero load latency.

performance with two VCs is lower than some partially adaptive functions even though it offers higher path diversity. For example, with bit reverse traffic, negative-first and odd-even show 47.1% and 30.6% higher performance than Duato, respectively. With two VCs, improving the VC utilization greatly outweighs the negative effect of HoL blocking induced by aggressive VC re-allocation.

Duato's performance increases significantly with four VCs. More VCs improve the possibility of facing empty VCs; the difference between aggressive and conservative VC re-allocation declines. For example, Duato achieves the highest performance for bit reverse. Most partially adaptive functions get significant improvement when the VC count increases from one to two, but only gain slightly when the VC count increases to four. More VCs mitigate the negative effect of HoL blocking. On the other hand, path diversity is an important factor. Configuring two VCs mitigates the HoL blocking, and it is also enough to stress the physical channel with aggressive VC re-allocation, making path diversity the performance bottleneck.

Duato's performance steadily increases with six VCs. Six VCs help to stress the physical channel with conservative VC re-allocation. Also, Duato provides enough path diversity, preventing it from rapidly becoming the bottleneck. The performance gain when the VC count increases from six to eight is small; with eight VCs, the physical path congestion becomes the limiting factor. A noteworthy phenomenon is observed for bit complement with eight VCs: although Duato has twice the path diversity of the three turn models, its performance is only slightly better. Most bit complement traffic traverses the center area of mesh, and Duato's high path diversity can only slightly mitigate this region congestion problem.

In summary, our evaluation of routing functions gives the following insights: 1. With limited VCs, providing efficient VC utilization greatly outweighs the negative effect of HoL blocking. The efficient VC utilization provided by aggressive VC re-allocation compensates for the limited path diversity of partially adaptive functions, resulting in higher performance than fully adaptive ones. 2. With more VCs, the offered path diversity of routing functions becomes the dominating factor for performance. 3. Configuring the appropriate VC count for NoCs must consider the applied routing function. For partially adaptive routing functions with aggressive VC re-allocation, generally two VCs are enough to provide high performance. More VCs not only add hardware overhead, but do not improve performance much. For the fully adaptive routing functions, at most eight VCs are enough for performance gains.

6.2 Single Region Performance

From previous analysis, Duato achieves the highest performance with more than six VCs; we choose it as

the routing function for the selection strategy analysis. Synthetic traffic evaluation is conducted with eight VCs, which are enough to stress the physical channel. We first evaluate the performance of two single region configurations: 4×4 and 8×8 meshes. There is only one traffic pattern throughout the whole network.

Synthetic Traffic Results. Figs. 14 and 15 give the latency results for transpose-1, bit reverse, shuffle and bit complement traffic patterns in 4×4 and 8×8 meshes, respectively. In the 4×4 mesh, DBSS has the best performance on these four traffic patterns as RCA suffers from intra-region interference. There is one exception: for bit complement, RCA's saturation point is 2.1% higher. Bit complement has the largest AHP with 4 hops in a 4×4 network (Fig. 3); this AHP mitigates the intra-region interference. LOCAL and NoP perform the worst for bit complement traffic due to their limited knowledge. The small AHP (2.5 hops) of transpose-1 leads to RCA performing the worst among all four adaptive algorithms. DBSS, LOCAL and NoP offer similar performance for transpose-1 with $\sim 13\%$ improvement in saturation throughput versus RCA. DBSS has a significant improvement of 21.9% relative to RCA for bit reverse.

DBSS improves saturation throughput by 10.2% and 8.5% over LOCAL for shuffle and bit complement traffic. These patterns cause global congestion and the shortsightedness of the locally adaptive strategy makes it unable to avoid congested areas. The saturation throughput improvements of DBSS over NoP are 17.7% and 11.1% for bit reverse and bit complement traffic respectively. NoP overlooks the status of neighbors. Comparing the performance of LOCAL against NoP further illuminates this limitation. This phenomenon validates our weighting mechanism placing more emphasis on closer nodes.

LOCAL outperforms RCA on a 4×4 mesh; intra-region interference leads RCA to make inferior decisions. However, in the 8×8 mesh, DBSS and RCA offer the best performance, while LOCAL has inferior performance. RCA's improvement comes from the weighting mechanism in the congestion propagation network. The weight of the congestion information halves for each hop; the effect of intra-region interference from distant nodes diminishes. This interference reduction is a result of the high AHP of 5.58 for these patterns. However, the AHP on the 4×4 network is 2.63, which is not large enough to hide the negative effect of interference. Although the weighted aggregation mechanism mitigates some interference in the 8×8 mesh, DBSS still outperforms RCA by 11.1% for bit reverse traffic. Compared with the 4×4 network, DBSS further improves the saturation throughput for shuffle and bit complement versus LOCAL by 12.4% and 16.5%. The shortsightedness of LOCAL has a stronger impact in a larger network. Similar trends are seen for NoP. For most traffic, DOR's rigidity prevents

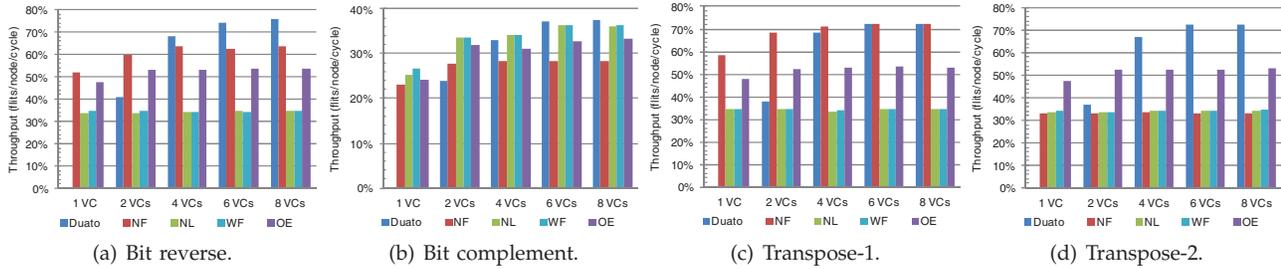


Fig. 13. The saturation throughput of several routing functions in a 4×4 mesh network.

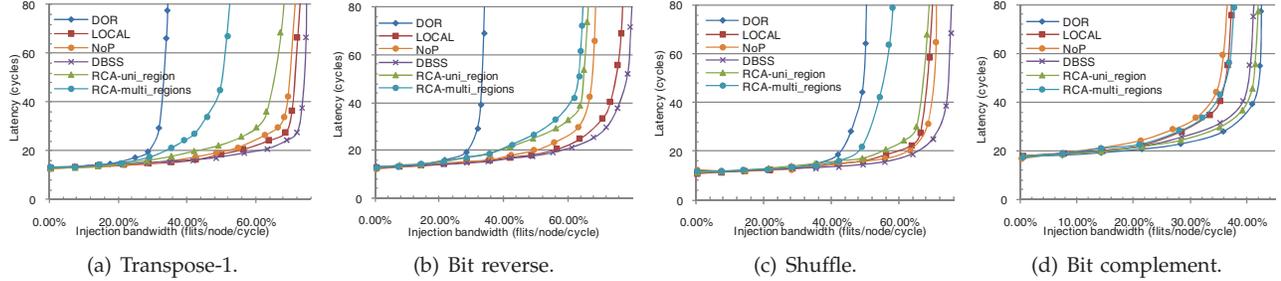


Fig. 14. Routing algorithm performance for a 4×4 mesh network (region). *RCA-uni_region* and *RCA-multi_region* give the performance of RCA for a single region and multiple regions respectively.

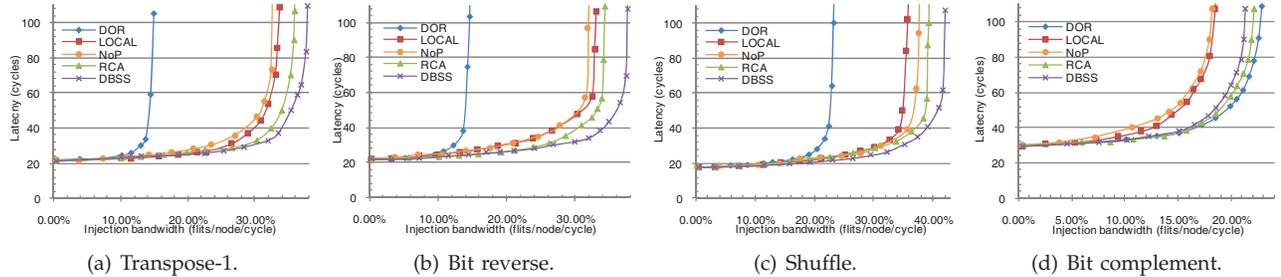


Fig. 15. Routing algorithm performance for an 8×8 mesh network with a single region.

it from avoiding congestion, resulting in the poorest performance.

Application Results. Fig. 16 shows the speedups relative to DOR for eight PARSEC workloads. The workloads can be classified into two groups: network-insensitive applications and network-sensitive applications. Sophisticated routing algorithms can improve the network saturation throughput, but the full-system performance improvements depend on the load and traffic pattern created by each application [48]. Applications with a high network load and significant bursty traffic receive benefit from advanced routing algorithms. For *blackscholes*, *canneal*, *raytrace* and *swaption*, their working sets fit into the private L2 caches, resulting in low network load. Different routing algorithms have similar performance for this group. The other four applications have lots of bursty traffic, emphasizing network-level optimization techniques; their performance benefits from routing algorithms supporting higher saturation throughput. For these four applications, DBSS and LOCAL get the best performance due to the small network size. DBSS achieves an average speedup of 9.6% and maximum 12.5% speedup over DOR. RCA performs better than NoP. NoP is worse than DOR for

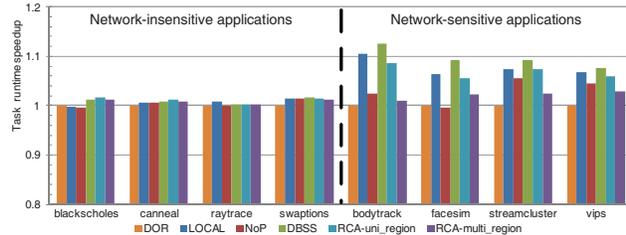


Fig. 16. System speedup for PARSEC workloads.

facesim. Its ignorance of neighboring nodes results in sub-optimal selections.

6.3 Multiple Region Performance

We evaluate two multiple-region configurations: one regular (Fig. 4) and one irregular region configuration (Fig. 17(a)). In both configurations, we focus on the performance of R0.

Small-Sized Regular Region Results. In this configuration (Fig. 4), regions R1-R3 run uniform random traffic with 4% injection rates while we vary the pattern in R0. LOCAL, NoP and DBSS do not have inter-region interference, since they only consider the congestion status of nodes belonging to the same region when making selections. Thus, in this configuration, all algorithms except RCA, have

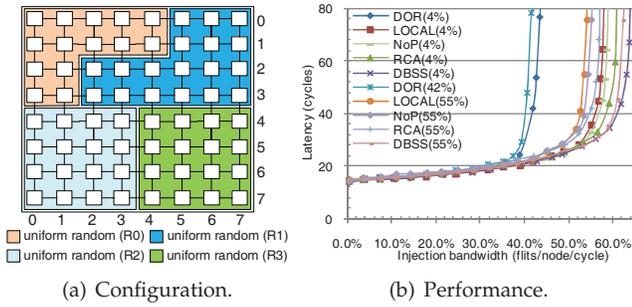


Fig. 17. Irregular region configuration and its performance.

the same performance as in Fig. 14. RCA’s performance suffers from inter-region interference. The ‘RCA-multi_regions’ curves in Fig. 14 show RCA’s performance for the multiple region configuration.

Compared to a single region, RCA’s performance declines; RCA suffers not only from intra-region interference, but also from inter-region interference. Transpose-1 and shuffle see 22.7% and 16.9% drops in saturation throughput. For bit reverse, the performance degradation is minor; the intra-region interference has already significantly degraded RCA’s performance and hides the effect of inter-region interference. DBSS maintains its performance for this configuration, thus revealing a clear advantage. The average saturation throughput improvement is 25.2% with the maximum improvement of 46.1% for transpose-1 traffic. Fig. 16 (‘RCA-multi_regions’) shows that RCA’s performance decreases for these four network-sensitive applications compared to the single region configuration (R1-R3 run uniform random traffic with 4% injection rates). *Bodytrack* has the maximum performance decrease of 7.3%.

Routers at the region boundaries strongly affect R0’s performance, since some of their input ports are never used. For example, the west input VCs of router (0,4) are always available since no packets arrive from the west. The interference from internal nodes of R1 and R2 is partially masked by RCA’s weighting mechanism at these boundary nodes with 8 free VCs. This explains why R0’s saturation point only decreases from 50% to 47% when the injection rate of R1 increases from 4% to 64% in Fig. 5.

Irregular Region Results. Fig. 17(a) shows non-rectangular regions. The isolation boundaries of R0 and R1 are the minimal rectangles surrounding them; traffic from both regions share some network links, so the traffic from R1 affects routing in R0 in some cases. Fig. 17(b) shows the performance of R0 while varying the injection rate of R1 from low load (4%) to high load (55%); the injection rates of R2 and R3 are fixed at 4%. All regions run uniform random traffic.

For both high and low loads in R1, DBSS has the best performance. As the load in R1 increases, the performance of all algorithms declines. For low load in R1, RCA has the second highest saturation throughput. Two rows of R0 have 5 routers; LOCAL

TABLE 2. Average throughput improvement of DBSS.

network	LOCAL	NoP	RCA	RCA_multi
4 × 4	7.2%	8.8%	10.4%	25.2%
8 × 8	12.6%	14.9%	4.7%	-
irregular	16.5%	14.3%	-	6.8%

and NoP are not sufficient to avoid congestion. When R1 has a high injection rate, the saturation points decline for DOR, LOCAL, NoP, RCA and DBSS by 7%, 7%, 6.8%, 6.7% and 4% respectively; DBSS has the least performance degradation since it offers the best isolation between these two regions. As these two regions are not completely isolated, some traffic for R0 and R1 share common network links; in this case, traffic from R1 should affect routing in R0. Only traffic between a subset of routers is completely isolated; DBSS correctly accounts for this dependency across the regions which makes it a flexible and powerful technique.

Summary. In workload consolidation scenarios, different applications will be mapped to different region sizes according to their intrinsic parallelism. However, with small regions, RCA suffers from interference, while LOCAL and NoP are limited by shortsightedness for large regions. Table 2 lists the average saturation throughput improvement of DBSS against other strategies. DBSS provides the best performance for all evaluated configurations and it shows the smallest performance degradation in multiple irregular regions. DBSS can provide more predictable performance when running multiple applications. DBSS is well suited to workload consolidation.

6.4 Concentrated Mesh Evaluation

Configuration. Here, we extend the analysis to the concentrated mesh (CMesh) topology [2], [32]. As a case study, we use radix-4 CMeshes [2], [32]; four cores are concentrated around one router, with two cores in each dimension, as shown in Fig. 18(a). Here, Core0, Core1, Core4 and Core5 are concentrated on router (0,0). Each core has its own injection/ejection channels to the router. Based on a CMesh latency model, the network channel has a 2-cycle delay, while the injection/ejection channel has 1 cycle delay [32]. The router pipeline is the same as discussed in Sec. 4.2. As shown in Fig. 18, we evaluate 16- and 64-core platforms. For the 64-core platform, both single- and multiple-region experiments are conducted. For multiple-region configuration (Fig. 18(b)), regions R1-R3 run uniform random traffic with 4% injection rate while we vary the pattern in R0.

Performance. Fig. 19 shows the results. In the 2×2 CMesh, LOCAL, DBSS and RCA have the same performance as there are only two routers along each dimension; these selection strategies utilize the same congestion information. For example, in Fig. 18(a),

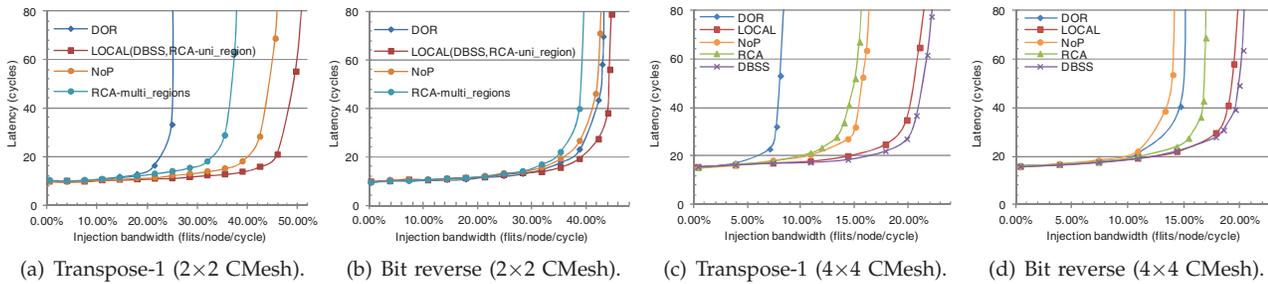


Fig. 19. Routing algorithm performance for CMeshes. *RCA-uni_region* and *RCA-multi_region* give the performance of RCA for a single region and multiple regions respectively.

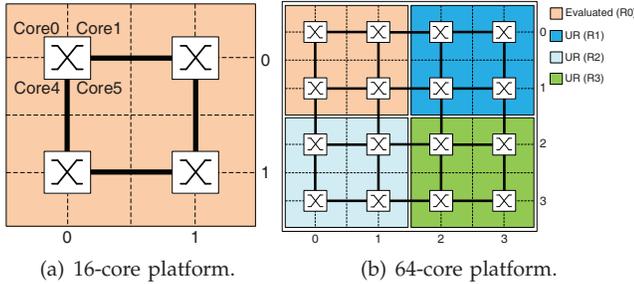


Fig. 18. The CMesh configurations.

when a packet is routed from router (0,1) to (1,0), LOCAL, DBSS and RCA all make selections by comparing the congestion status of the east input port of (0,0) and the north input port of (1,1). NoP performs differently; it compares the status of the north and south input ports of (1,0). This difference results in poorer performance for NoP (Figs. 19(a) and 19(b)), since it ignores the status of the nearest nodes. DOR has the lowest performance for transpose-1 (Fig. 19(a)). With bit reverse in a 4×4 mesh, DOR's performance is about one third of the performance of the best adaptive algorithm (Fig. 14(a)). But in a 2×2 CMesh with bit reverse, the performance difference between DOR and adaptive routing declines (Fig. 19(b)). Concentration reduces the average hop count; adaptive routing has less opportunity to balance network load.

In the multiple-region scenario of a 4×4 CMesh, RCA's performance drops due to the inter-region interference. There is a 26.2% performance decline with transpose-1 (*RCA-uni_region* vs. *RCA-multi_region* in Fig. 19(a)). In a 4×4 mesh region with bit reverse, the inter-region interference only slightly affects RCA since the intra-region interference already strongly deteriorates its performance (Fig. 14(b)). However, here with bit reverse, the inter-region interference brings a 9.8% performance drop as there is no intra-region interference in a 2×2 CMesh (Fig. 19(b)).

With one region configured in a 4×4 CMesh, the performance trend (Figs. 19(c) and 19(d)) is generally similar to that of a 4×4 mesh (Figs. 14(a) and 14(b)); DBSS and LOCAL show the highest performance while the RCA suffers from intra-region interference. RCA's performance is 29.4% lower than that of DBSS in a 4×4 CMesh (Fig. 19(c)), while this performance gap is only 7.4% in a 4×4 mesh (Fig. 14(a)). The

effect of intra-region interference increases due to the change of the latency ratio between the network and injection/ejection channel, resulting in making judicious selection become more important in a 4×4 CMesh. These results indicate that considering the appropriate congestion information is also important in CMeshes; DBSS still offers high performance.

7 HARDWARE OVERHEAD

Wiring Overhead. Adaptive routers require some wiring overhead to transmit congestion information. Assuming an 8×8 mesh with 8 VCs/port, DBSS introduces 8 additional wires for each dimension. RCA uses 8 wires in each direction for a total of 16 per dimension. NoP requires $4 \times \log(\text{numVCs}) = 12$ wires per direction; there are 24 wires for one dimension. LOCAL requires $\log(\text{numVCs}) = 3$ wires in each direction for 6 total wires per dimension. Given a state-of-the-art NoC [23] with 128-bit flit channels, the overhead of DBSS is just 3.125% versus 6.25%, 9.375% and 2.34% for RCA, NoP and LOCAL, respectively. DBSS has a modest overhead; abundant wiring on chip is able to accommodate these wires.

Router Overhead. DBSS adds the SMC, DP and three registers to the canonical router. In an 8×8 mesh network, SMC utilizes two 7-bit shifters to select and align the congestion information of the two dimensions. The 64-to-1 multiplexer of DP can be implemented in tree format. *Congestion_X* and *congestion_Y* are 9 bits wide, while *out_dim* is 64 bits wide. Considering about the wide (128-bit) datapath of our mesh network (5 ports, 8 VCs/port and 5 slots/VC), the overhead of these registers is only 0.3% compared to the existing buffers.

Power Consumption. We leverage an existing NoC power model [40], which divides the total power consumption into three main components: channels, input buffers and router control logics. We also model the power consumption of the congestion propagation network. The activity of these components is obtained from BookSim. We use a 32nm technology process with a 1 GHz clock frequency.

Fig. 20(a) shows the average power for transpose-1 traffic in an 8×8 mesh. Since DOR cannot support injection rates higher than 20%, there are no results for

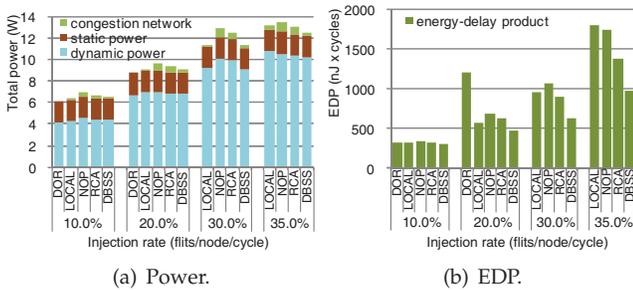


Fig. 20. Power consumption results for transpose-1 traffic.

30% and 35% injection rates. The increased hardware complexity, especially the congestion propagation network of adaptive routing results in a higher average power than the simple DOR. Comparing these adaptive routers, LOCAL and DBSS have the lowest power due to their lowest wiring overhead. NoP has the highest power. LOCAL needs less additional wires than DBSS, but these wires have a higher activity factor. For a 20% injection rate, the activity of DBSS’s congestion propagation network is 15.8% versus 17.5% of LOCAL. This smaller activity factor mitigates the increased power of DBSS’s congestion propagation network. For a 35% injection rate, LOCAL consumes more power than DBSS. The adaptive routing accelerates packet transmission, showing a significant energy-delay product (EDP) advantage. As shown in Fig. 20(b), DBSS provides smallest EDP for medium (20%) and high injection rates (30% and 35%).

8 IN-DEPTH ANALYSIS OF INTERFERENCE

Here, we delve into the cause of interference. We find that it is strongly related to the detailed implementation of the congestion metric. There are many metrics, such as free VCs, free buffers and crossbar demands [22]. We use free buffers as an example.

There are two implementation choices. One is to choose the output port with the *maximal free buffers*, and the other is to use *minimal occupied buffers*. These two choices appear equivalent at the first glance. Indeed, for the local selection strategy, they are the same. This is not the case for RCA due to the interference. To clarify this point, consider the packet routing example shown in Fig. 2(c). The following equation shows the calculation of the congestion metrics:

$$\begin{cases} M_w = W(0, 1) + 0.5 \times W(0, 0) \\ M_s = S(1, 2) + 0.5 \times S(2, 2) + 0.25 \times S(3, 2) \end{cases} \quad (1)$$

In Eqn. 1, M_w and M_s are the congestion metrics for the west and south output ports. The $W(i, j)$ and $S(i, j)$ are the free buffer count (or the occupied buffer count according to the implementation choice) in the west and south input ports of router (i, j) . The coefficients, such as the 0.5 and 0.25, are due to the weighting mechanism used by RCA.

Since fully adaptive routing uses the conservative VC re-allocation scheme, and the packet length in

NoCs is generally short, most of the buffer slots are unoccupied. Let us consider that an example range of occupied buffers $occupied_{range}$ for each input port is $0 \leq occupied_{range} \leq 6$. Then the range of free buffers $free_{range}$ for each input port is $34 \leq free_{range} \leq 40$ in our experimental configuration. If we implement the *minimal occupied buffers* as the congestion metric, then the range of M_w in Eqn. 1 is $0 \leq M_w \leq 9$. Similarly, the range of M_s in Eqn. 1 is $0 \leq M_s \leq 10.5$. The ranges of M_w and M_s are nearly the same.

However, if we use the *maximal free buffers* as the congestion metric, then the range of M_w in Eqn. 1 is $51 \leq M_w \leq 60$. Similarly, the range of M_s is $59.5 \leq M_s \leq 70$. Unlike the situation with *minimal occupied buffers*, the ranges of M_w and M_s have almost no overlap with the *maximal free buffers* implementation choice. For almost all situations, M_s is larger than M_w ; this packet loses significant adaptivity as RCA will always choose the south port as the optimal one.

This example shows that the interference is strongly related to the detailed implementation choices of the same congestion metric. We find that if the output port with the *maximal free buffers* is chosen as the optimal one in RCA, its saturation throughput is about 1/2 of LOCAL for most synthetic traffic patterns. But if the output port with the *minimal occupied buffers* is chosen as the optimal one, RCA’s performance is almost the same as LOCAL.

Even with aggressive VC re-allocation in partially adaptive routing, *maximal free buffers* and *minimal occupied buffers* show similar properties as well. The reason is that network saturates when some resources are saturated [13]. In particular, these central buffers, VCs and links may easily saturate while other resources may have low utilization [42]. Kim et al. observed that even with one VC per physical channel, the average buffer utilization is below 40% at saturation [29]; more resources are in free status than in occupied status. Due to this reason, other metrics, such as ‘free VCs’, have similar properties; the ‘maximum free VCs’ implementation used in previous evaluation, has inferior performance compared to the ‘minimal occupied VCs’ metric. Although RCA achieves good performance by carefully implementing the congestion metric, its instability with different implementation choices easily leads to an inferior design. However, DBSS shows stable performance when either maximal free buffers (VCs) or minimal occupied buffers (VCs) since it eliminates the interference.

9 DBSS DESIGN SPACE EXPLORATION

Number of Propagation Wires. We evaluate the saturation throughput of DBSS with 1-, 2- and 3-bit wide propagation networks. Ma et al. [36] shows the detailed results. The increase in wiring yields only minor performance improvements, and these performance gains decrease as the network scales.

Making a fine distinction about the available VCs has little practical impact as the selection strategy is only interested in the relative difference between two routing candidates. To demonstrate this point, we simultaneously keep the 1-, 2-, 3-bit status information and record the fraction of times the selection strategy using 1 bit or 2 bits makes a different selection than the 3-bit status in a 4×4 mesh. At very low network load (2% injection rate) with uniform random traffic, the 1- and 2-bit status makes a different selection 8.8% and 7.6% of the time. This difference mainly comes from the randomization when facing two equal output port statuses. These fractions decrease with higher network loads. The fractions are 1.8% and 1.1%, respectively, at saturation. This minor difference verifies that making fine distinction is not necessary.

DBSS Scalability. The cost of scaling DBSS to a larger network increases linearly as N 1-bit congestion propagation wires are needed for each row (column) in an $N \times N$ network. For a 16×16 mesh, this represents a 6.25% overhead with 128-bit flit channels. The size of the added registers in Fig. 9 increases linearly. The latency of DP module increases logarithmically with network radix; however this delay is not on the critical path so it will not increase the cycle time. DBSS is a cost effective solution for many-core networks.

Congestion Propagation Delay. In addition to eliminating interference, our novel congestion network operates with only a 1 cycle/hop delay compared to 2 cycles/hop in RCA. To isolate this effect from interference effects, we compare DBSS with a 1 cycle/hop and a 2 cycles/hop congestion propagation network. The timeliness of the 1 cycle per hop network improves saturation throughput by up to 5% over the 2-cycle design (for shuffle traffic).

10 CONCLUSION

The shortsightedness of locally adaptive routing limits the performance for large-sized networks, while globally adaptive routing suffers from interference for multiple regions. Interference (or false dependency) comes from utilizing network status across region boundaries. By leveraging a novel congestion propagation network, DBSS provides both high adaptivity for congestion avoidance and dynamic isolation to eliminate interference. Although DBSS does not provide strict isolation, it is a powerful technique that can dynamically adapt to changing region configurations as threads are migrated or rescheduled. Experimental results show that DBSS offers high performance in all evaluated network configurations. The wiring overhead of DBSS is small. We provide additional insights into the design of routing functions.

ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for helpful suggestions. This work is supported by the University

of Toronto, NSERC, the Connaught Fund, 863 Program of China(2012AA010905), NSFC(61070037, 61025009, 60903039, 61103016), China Edu. Fund.(20094307120012), Hunan Prov. Innov. Fund. For PostGrad.(CX2010B032).

REFERENCES

- [1] G. Ascia, V. Catania, M. Palesi, and D. Patti. Implementation and analysis of a new selection strategy for adaptive routing in networks-on-chip. *Computers, IEEE Transactions on*, 57(6):809–820, June 2008.
- [2] J. Balfour and W. Dally. Design tradeoffs for tiled CMP on-chip networks. In *ICS*, 2006.
- [3] S. Bell et al. TILE64 - processor: A 64-core SoC with mesh interconnect. In *ISSCC*, pages 88–598, February 2008.
- [4] C. Bienia, S. Kumar, J. P. Singh, and K. Li. The PARSEC benchmark suite: characterization and architectural implications. In *PACT*, 2008.
- [5] S. Boyd-Wickizer, A. T. Clements, Y. Mao, A. Pesterev, M. F. Kaashoek, R. Morris, and N. Zeldovich. An analysis of Linux scalability to many cores. In *OSDI*, 2010.
- [6] G.-M. Chiu. The odd-even turn model for adaptive routing. *Parallel and Distributed Systems, IEEE Transactions on*, 11(7):729–738, July 2000.
- [7] C.-L. Chou and R. Marculescu. Run-time task allocation considering user behavior in embedded multiprocessor networks-on-chip. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 29(1):78–91, January 2010.
- [8] C.-L. Chou, U. Ogras, and R. Marculescu. Energy- and performance-aware incremental mapping for networks on chip with multiple voltage levels. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 27(10):1866–1879, October 2008.
- [9] W. Dally. Virtual-channel flow control. *Parallel and Distributed Systems, IEEE Transactions on*, 3(2):194–205, March 1992.
- [10] W. Dally and H. Aoki. Deadlock-free adaptive routing in multicomputer networks using virtual channels. *Parallel and Distributed Systems, IEEE Transactions on*, 4:466–475, April 1993.
- [11] W. Dally and C. Seitz. Deadlock-free message routing in multiprocessor interconnection networks. *Computers, IEEE Transactions on*, C-36(5):547–553, May 1987.
- [12] W. Dally and B. Towles. Route packets, not wires: on-chip interconnection networks. In *DAC*, pages 684–689, May 2001.
- [13] W. Dally and B. Towles. *Principles and Practices of Interconnection Networks*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2003.
- [14] R. Das, A. Mishra, C. Nicopoulos, D. Park, V. Narayanan, R. Iyer, M. Yousif, and C. Das. Performance and power optimization through data compression in network-on-chip architectures. In *HPCA*, 2008.
- [15] J. Duato. A new theory of deadlock-free adaptive routing in wormhole networks. *Parallel and Distributed Systems, IEEE Transactions on*, 4(12):1320–1331, December 1993.
- [16] J. Duato. A necessary and sufficient condition for deadlock-free adaptive routing in wormhole networks. *Parallel and Distributed Systems, IEEE Trans. on*, 6(10):1055–1067, Oct. 1995.
- [17] N. Enright Jerger and L. Peh. *On-Chip Networks*. Morgan and Claypool Publishers, San Francisco, CA, USA, 1 edition, 2009.
- [18] W.-C. Feng and K. G. Shin. Impact of selection functions on routing algorithm performance in multicomputer networks. In *ICS*, pages 132–139, July 1997.
- [19] B. Fu, Y. Han, J. Ma, H. Li, and X. Li. An abacus turn model for time/space-efficient reconfigurable routing. In *ISCA*, 2011.
- [20] M. Gallet. Spider: a high-speed network interconnect. *Micro, IEEE*, 17(1):34–39, January-February 1997.
- [21] C. Glass and L. Ni. The turn model for adaptive routing. In *ISCA 1992*, pages 278–287, June 1992.
- [22] P. Gratz, B. Grot, and S. Keckler. Regional congestion awareness for load balance in networks-on-chip. In *HPCA*, 2008.
- [23] P. Gratz, C. Kim, K. Sankaralingam, H. Hanson, P. Shivakumar, S. Keckler, and D. Burger. On-chip interconnection networks of the TRIPS chip. *Micro, IEEE*, 27(5):41–50, Sept.-Oct. 2007.
- [24] P. Gratz, K. Sankaralingam, H. Hanson, P. Shivakumar, R. McDonald, S. Keckler, and D. Burger. Implementation and evaluation of a dynamically routed processor operand network. In *NOCS*, pages 7–17, May 2007.

[25] Y. Hoskote, S. Vangal, A. Singh, N. Borkar, and S. Borkar. A 5-GHz mesh interconnect for a Teraflops processor. *Micro, IEEE*, 27(5):51–61, Sept.-Oct. 2007.

[26] J. Hu and R. Marculescu. DyAD - smart routing for networks-on-chip. In *DAC*, pages 260–263, June 2004.

[27] J. Hu and R. Marculescu. Energy- and performance-aware mapping for regular NoC architectures. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Trans. on*, April 2005.

[28] M. Karol, M. Hluchyj, and S. Morgan. Input versus output queueing on a space-division packet switch. *Communications, IEEE Transactions on*, 35(12):1347–1356, December 1987.

[29] G. Kim, J. Kim, and S. Yoo. Flexibuffer: Reducing leakage power in on-chip network routers. In *DAC*, June 2011.

[30] J. Kim, D. Park, T. Theocharides, N. Vijaykrishnan, and C. Das. A low latency router supporting adaptivity for on-chip interconnects. In *DAC*, pages 559–564, 2005.

[31] A. Kumar, P. Kundu, A. Singh, L.-S. Peh, and N. Jha. A 4.6Tbits/s 3.6GHz single-cycle NoC router with a novel switch allocator in 65nm CMOS. In *ICCD*, pages 63–70, Oct. 2007.

[32] P. Kumar, Y. Pan, J. Kim, G. Memik, and A. Choudhary. Exploring concentration and channel slicing in on-chip network router. In *NOCS*, 2009.

[33] T. Lei and S. Kumar. A two-step genetic algorithm for mapping task graphs to a network on chip architecture. In *DSD*, pages 180–187, September 2003.

[34] M. Li, Q.-A. Zeng, and W.-B. Jone. DyXY - a proximity congestion-aware deadlock-free dynamic routing method for network on chip. In *DAC*, pages 849–852, June 2006.

[35] D. Iltzky, J. Hoffman, A. Chun, and B. Esparza. Architecture of the scalable communications core’s network on chip. *Micro, IEEE*, 27(5):62–74, September-October 2007.

[36] S. Ma, N. Enright Jerger, and Z. Wang. DBAR: an efficient routing algorithm to support multiple concurrent applications in networks-on-chip. In *ISCA*, 2011.

[37] S. Ma, N. Enright Jerger, and Z. Wang. Whole packet forwarding: Efficient design of fully adaptive routing algorithms for networks-on-chip. In *HPCA*, 2012.

[38] P. S. Magnusson, M. Christensson, J. Eskilson, D. Forsgren, G. Hällberg, J. Högberg, F. Larsson, A. Moestedt, and B. Werner. Simics: A full system simulation platform. *Computer*, 35:50–58, February 2002.

[39] J. C. Martínez, F. Silla, P. López, and J. Duato. On the influence of the selection function on the performance of networks of workstations. In *ISHPC*, pages 292–299, October 2000.

[40] G. Michelogiannakis, D. Sanchez, W. Dally, and C. Kozyrakis. Evaluating bufferless flow control for on-chip networks. In *NOCS*, pages 9–16, May 2010.

[41] M. Mirza-Aghatabar et al. An empirical investigation of mesh and torus NoC topologies under different routing algorithms and traffic models. In *DSD*, pages 19–26, 2007.

[42] A. K. Mishra, N. Vijaykrishnan, and C. R. Das. A case for heterogeneous on-chip interconnects for cmps. In *ISCA*, 2011.

[43] O. Mutlu and T. Moscibroda. Parallelism-aware batch scheduling: Enhancing both performance and fairness of shared DRAM systems. In *ISCA*, pages 63–74, June 2008.

[44] N. Neelakantam, C. Blundell, J. Devietti, M. M. Martin, and C. Zilles. FeS2: A full-system execution-driven simulator for x86. In *Poster presented at ASPLOS*, 2008.

[45] L. Peh and W. Dally. A delay model and speculative architecture for pipelined routers. In *HPCA*, 2001.

[46] R. S. Ramanujam and B. Lin. Destination-based adaptive routing on 2D mesh networks. In *ANCS*, 2010.

[47] S. Rodrigo, J. Flich, J. Duato, and M. Hummel. Efficient unicast and multicast support for CMPs. In *MICRO*, Nov 2008.

[48] D. Sanchez, G. Michelogiannakis, and C. Kozyrakis. An analysis of on-chip interconnection networks for large-scale chip multiprocessors. *ACM Trans. Archit. Code Optim.*, 7(1):4:1–4:28, May 2010.

[49] L. Schwiebert and R. Bell. Performance tuning of adaptive wormhole routing through selection function choice. *J. Parallel Distrib. Comput.*, 62:1121–1141, July 2002.

[50] S. Zhuravlev, S. Blagodurov, and A. Fedorova. Addressing shared resource contention in multicore processors via scheduling. In *ASPLOS*, pages 129–142, March 2010.



Sheng Ma received the B.S. degree in computer science from National University of Defense Technology (NUDT) in 2007. Since Feb. 2009, he has been a PhD candidate of NUDT. Since Sept. 2010, he has been visiting the University of Toronto. His research interests include on-chip networks, SIMD architectures and arithmetic unit designs. He is a student member of the IEEE.



Natalie Enright Jerger received the B.A.Sc. degree from the Department of Electrical and Computer Engineering at Purdue University, and the M.S.E.E. and Ph.D. degrees from the Department of Electrical and Computer Engineering, University of Wisconsin - Madison. She is currently an Assistant Professor in the Electrical and Computer Engineering Department at the University of Toronto. Her research interests include on-chip networks, many-core architectures and cache coherence protocols. She is a member of the IEEE and ACM.



Zhiying Wang received the PhD degree in electrical engineering from NUDT in 1988. He is currently a Professor with School of Computer, NUDT. He has contributed over 10 invited chapters to book volumes, published 240 papers in archival journals and refereed conference proceedings, and delivered over 30 keynotes. His main research fields include computer architecture, computer security, VLSI design, reliable architecture, multi-core memory system and asynchronous circuit. He is a member of the IEEE and ACM.



Mingche Lai received the PhD degree in computer engineering from NUDT in 2008. Currently, he is an Associate Professor with School of Computer, NUDT, and employed to develop high-performance computer interconnection systems. Since 2008, he has also been a Faculty Member with National Key Laboratory for Parallel and Distributed Processing of China. His research interests include on-chip networks, optical communication, many-core processor architecture, hardware/software co-design. He is a member of the IEEE and ACM.



Libo Huang received the BS and PhD degree in computer engineering from NUDT in 2005 and 2010 respectively. Currently, he is an Assistant Professor with School of Computer, NUDT. His research interests include computer architecture, hardware/software Co-design, VLSI design, on-chip communication. He served as the technical reviewer of several conferences and journals, e.g., MEJ, IJHPSA, ICCE, CJ. Since 2004, he authored more than 20 papers in internationally recognized journals and conferences.