The Effect of Virtual Channels on the Performance of Wormhole Algorithms in Multicomputer Networks

Wu-chang Feng
Advisor: Kang Shin

Real-Time Computing Laboratory
Department of Electrical Engineering and Computer Science
The University of Michigan
Ann Arbor, MI 48109-2122

The work reported in this paper was supported in part by the National Science Foundation under Grant MIP-9203895. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the view of the NSF.

The Effect of Virtual Channels on the Performance of Wormhole Algorithms in Multicomputer Networks

Abstract

In this study, we examine the effects that traffic patterns and virtual channels have on the performance of wormhole routing algorithms. In particular we look at two algorithms, dimension-ordered routing and an adaptive algorithm based on Duato's methodology. From simulations of these schemes, we first show that the adaptive algorithm needs fewer virtual channels than dimension-ordered routing before its performance plateaus. Then, we show that depending on the traffic pattern, dimension-ordered routing, supplemented with additional virtual channels, can approach that of the adaptive algorithm.

1 Introduction

The performance of message passing in multicomputers depends on the routing and switching schemes employed in their interconnection network [1]. First generation multicomputers have typically employed store-and-forward techniques [2] to pass messages from source to destination. Each time a message arrives at a node it is buffered and the host processor executes a routing algorithm to determine the next step of the message.

Because store-and-forward approaches incur both buffer and processing overheads at intermediate nodes, most second-generation commercial and research multicomputers use wormhole switching to route packets [3–7]. In wormhole switching, a message is split up into several flow control digits (flits) which are buffered at each node. As the header flit is routed through the network, the remaining flits follow directly behind it in a pipelined fashion. When the header flit can't be routed due to contention in the network, the flow control within the network blocks the message, effectively dilating its length until a link becomes free.

Wormhole switching presents a wide variety of problems which have been heavily studied in many places. Much of the work has been in the development and evaluation of new routing algorithms. While this is an important aspect to the design of such networks, there are many other parameters to consider. In this paper, we first present a general classification of the parameters involved in designing these networks. We then look at the relationship between a subset of these parameters and how they effect performance.

Section 2 describes the general issues in the design of wormhole networks and casts our work within it. Section 3 gives some background on the specific parameters studied in this paper. Section 4 describes the actual experimental parameters and the simulation environment. Section 5 evaluates the network performance as these parameters vary. Finally, Section 6 concludes the report and presents additional work to be done.

2 General Issues

Many papers have looked at particular aspects of routing and switching in multicomputer networks, each assuming a particular architecture or set of parameters. Each study can be classified as it relates to the following list of parameters.

Virtual Channels: Because wormhole packets can block in the network and thus occupy switch resources, they are susceptible to deadlocks. Virtual channels were first introduced in [8] in order to prevent deadlocks that can occur in wormhole networks. In [9], Dally shows that virtual channels can also be used to increase network throughput and latency by relieving contention. This is because blocked packets can be bypassed by other packets using separate virtual channels. There are many issues involved when studying virtual channels including the number of virtual channels used, how virtual channels are multiplexed on the physical link, and how virtual channels are allocated to the virtual subnetworks within a particular routing algorithm. Dally [9] has examined the first two issues with respect to static (dimension-ordered) routing. The third issue is has been relatively unexplored.

Routing Algorithms: Routing algorithms determine the path of a packet between two nodes. These algorithms can be minimal, non-minimal, static, partially adaptive or fully adaptive. Minimal algorithms consider only the shortest paths. Non-minimal algorithms consider paths of multiple lengths which are not necessarily the shortest paths. Static algorithms consider only one fixed path. Partially adaptive algorithms choose a subset of the possible paths. Finally, fully adaptive algorithms do not restrict the selection of paths between two nodes. Each algorithm has specific properties and requirements which must be considered when comparing them. For example, fully adaptive routing algorithms generally require more virtual channels than partially adaptive ones in order to prevent deadlocks. In [10], several adaptive algorithms are compared, however, in these experiments, each algorithm is given as many virtual channels as it requires for deadlock prevention. This puts partially adaptive schemes at a disadvantage compared to fully adaptive schemes with larger numbers of virtual channels. A study of how effectively routing algorithms use their virtual channels has not been studied and would be of interest.

Traffic Patterns: The sensitivity of routing algorithms to traffic patterns, especially non-uniform patterns, has been studied in many of the papers on adaptive routing. Traffic characteristics such as message length, message interarrival times at the sources, and destination distributions can have significant performance implications. In [11], the first issue is studied. Many of the papers on adaptive routing address the latter two issues in the generation of non-uniform traffic. Traffic patterns generated from real applications would be ideal in any evaluation, but to date, we know of only one study [12] which has examined this.

Topology: The behavior of routing algorithms over a wide range of topologies and sizes is an area that has also been understudied. The influence that the connectivity and the size of networks have on traffic characteristics such as contention would be interesting. Comparative experiments between a set of routing algorithms across a range of sizes and topologies may give insight as to the strengths and weaknesses of particular algorithm designs.

Switch Parameters: In most studies, the architecture of the switch itself has largely been fixed. For example, how the switch is internally connected, how arbitration for idle virtual channels is done, and how the host and contention for host resources are modelled are parameters which are usually fixed in each study. In places where these issues are addressed, it has been shown that they impact performance considerably. For example, in [10, 13], throttling injection into the network keeps injection traffic from hogging all of the virtual channels and thus has a positive influence on network performance.

Evaluation Metrics: Finally, the actual metrics used in the evaluation of the network is important when comparing algorithms. Comparisons should ideally look at all metrics including packet latency, variance, and worst-case delay along with network throughput.

The work done in this study focuses on how the number of virtual channels, the traffic pattern, and the routing algorithm impact the performance of the network. We classify our experiments in the above framework as follows:

Our work:

Routing Algorithm: 1) Dimension-ordered 2) Minimal, fully-adaptive algorithm based on Duato's methodology. Selection function based on minimum congestion.
Virtual Channels: 2,4,6,8
Virtual Channel Multiplexing: Random/Fair among competing virtual channels
Virtual Channel Allocation: 1) Dimension-ordered: distributed evenly across two subnetworks 2) Adaptive: additional channels allocated to adaptive subnetwork.
Network Size: 64
Topology: 8-ary 2-cube
Destination Distribution: Uniform and non-uniform
Message Generation Distribution: Poisson
Message Length: 64 flits
Switch Architecture: Based on SPIDER [14, 15]. Bus interconnect. Fair, random arbitration for output channels.
Evaluation Metrics: Latency

Note that we are specifically interested in how virtual channels, routing algorithms, and traffic patterns affect the performance of networks. Dally [9] has looked at similar effects using dimension-ordered routing and uniform traffic. Boppana [10] has compared the performance of various algorithms, but each algorithm is given just enough virtual channels to prevent deadlock. This work attempts to examine, simultaneously, the parameters both papers address individually.

To more fully characterize the related work that has been done, we have included in Appendix A a classification of a number of papers which examine these issues.

3 Routing Algorithms and Virtual Channels

Since blocked wormhole packets hold link resources, routing algorithms which use wormhole switching must be designed to avoid deadlocks. Deadlocks occur when no message can advance towards its destination due to filled queues. Dally [8] has proposed a methodology to design static routing algorithms using virtual channels to eliminate the cycles in the dependency graph that cause deadlocks. While this removes the possibility of deadlocks, having a restricted number of paths between two nodes reduces fault tolerance and can lead to a communication bottleneck.

To alleviate such problems, adaptive routing algorithms for various topologies have been proposed. Linder and Harden [16] and Berman *et al.* [17] have developed fully-adaptive minimal path routing algorithms for k-ary n-cubes using $(n+1)2^{n-1}$ and 10(n-1)+6 virtual channels, respectively. While these schemes provide the desired adaptivity, the cost in virtual channels as n increases can be high. Glass and Ni [18] with the turn model and Chien and Kim [19] with planar-adaptivity propose methodologies for designing partially-adaptive deadlock-free algorithms using a limited number of virtual channels. This allows for some adaptivity without the high costs associated with fully-adaptive schemes. Finally, Dally and Aoki [13] and Duato [20–22] present several fully adaptive non-minimal algorithms which allow cycles in the channel-dependency graphs, but can still avoid deadlock with a restricted number of virtual channels.

While adaptive algorithms provide better performance and added fault tolerance, static algorithms allow for faster, simpler, asynchronous switch designs [13] and do not suffer problems with non-uniformities in resource usage [23]. One design alternative that can be considered is implementing dimension-ordered routing with extra virtual channels instead of adaptive routing since virtual channels provide substantial performance improvement [9] and are relatively inexpensive compared to the logic involved in implementing adaptive routing. In [9], Dally shows the benefits that extra virtual channels provide with the proper flit and packet sequencing. He also shows that these benefits diminish significantly as the number of virtual channels increases. In his study, Dally uses dimension-ordered routing with uniform traffic patterns to study these effects. In our experiments, we examine the same effects using an adaptive wormhole routing algorithm and hot-spot traffic patterns and compare them to dimension-ordered routing and uniform traffic patterns. Through this comparison, we evaluate whether dimension-ordered routing using additional virtual channels can outperform adaptive routing.

4 Experiments

In our experiments, we examine the performance of a minimal-path fully adaptive algorithm and compare it to that of a dimension-ordered algorithm. All of these experiments were performed on an 8x8 wrapped square mesh (torus). Figure 1 shows an example of a 4x4 torus. To make dimension-ordered routing deadlock-free, at least two virtual channels are provided [8]. This partitions the network effectively into a high and a low subnetwork. Additional virtual channels are then mapped evenly across the two subnetworks so that both the high and the low virtual networks have equivalent numbers of virtual channels.

The adaptive routing algorithm we use is derived from Duato's methodology [20–22]. This

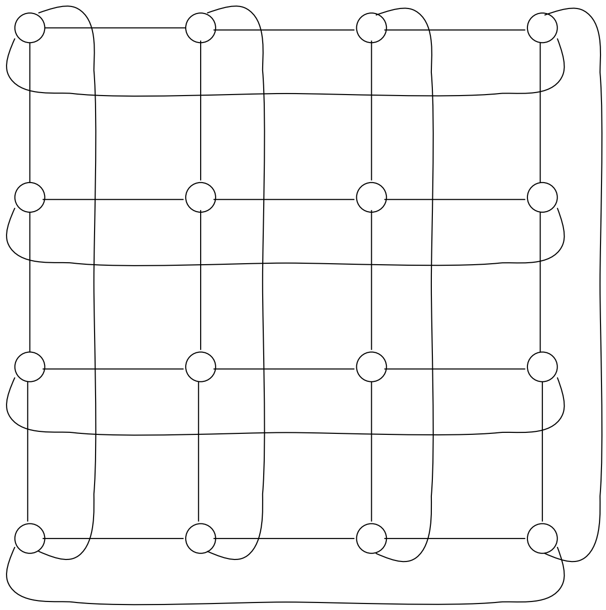


Figure 1: A 4x4 torus

methodology allows for the development of highly-adaptive algorithms without a large number of virtual channels. This algorithm partitions the network into two subnetworks: an adaptive, non-deterministic one and a static, deterministic one. Packets routing on the network are allowed to route down any free profitable virtual channel. When no free channel is found, the packet blocks and waits on a pre-specified channel in the static subnetwork. From then on, the packet is routed within this subnetwork using dimension-ordered routing until it reaches its destination. In our experiments, the static subnetwork always contained two virtual channels. Additional virtual channels were then used in the adaptive subnetwork.

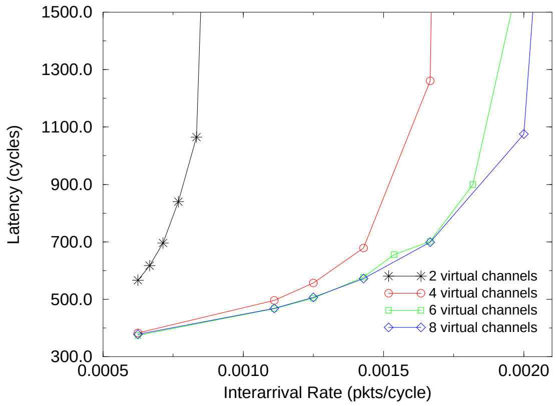
In order to evaluate these algorithms, pp-mess-sim, a network simulator [24], was used. pp-mess-sim provides a general framework for evaluating multicomputer router architectures. One of the features of pp-mess-sim is the ability to develop routing algorithms independently of the switch architecture model. This was done by defining a clear interface and information flow between the lower level switch model and the routing algorithm. Thus, any routing algorithm written for one switch model can be directly used on another switch model as long as the models adhere to the same protocol. For this study, two routing algorithms based on Duato's methodology were written, one of which was used in these simulations.

pp-mess-sim currently includes a cycle-level model of SPIDER [14,15], a communication adapter designed for HARTS [25]. Our simulations use this model as the basis for evaluating these algorithms. Some of the more important aspects of SPIDER include a demand-slotted bus interconnection between input and output channels and a fair arbitration scheme between virtual channels competing for a given physical link.

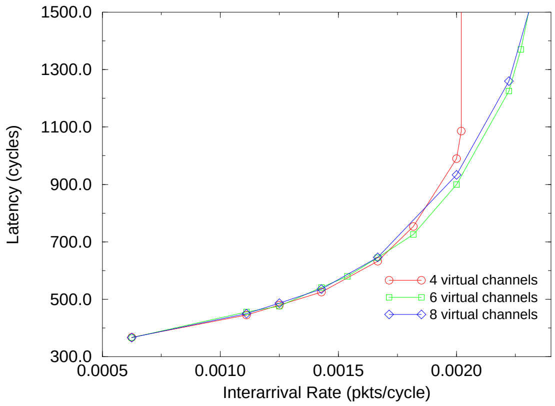
The experiments were run using 64-byte packets generated according to a Poisson arrival process whose parameter was varied across a range of values. Destinations for the packets were chosen uniformly across all other nodes in the network. For the hot-spot experiments, two nodes in the network were made "hot" by having their adjacent neighbors transmit additional packets. Each neighbor generates an additional 0.001 packets/cycle to send to the "hot" node.

5 Results

To see the effects of additional virtual channels on end-to-end packet latency we ran these experiments with a range of loads.

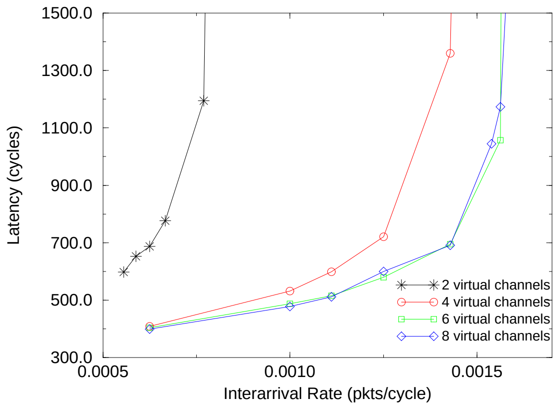


(a) Average latency for dimension-ordered routing

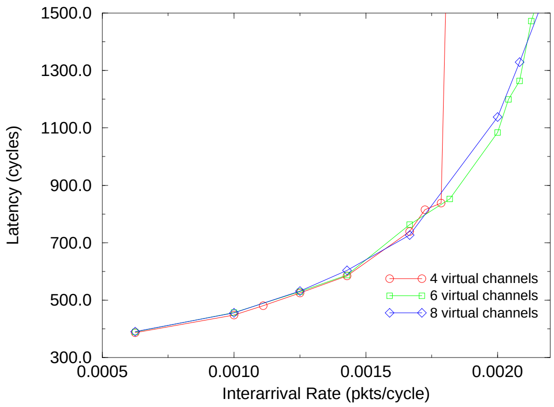


(b) Average latency for adaptive routing

Figure 2: Performance of wormhole algorithms under uniform traffic load.

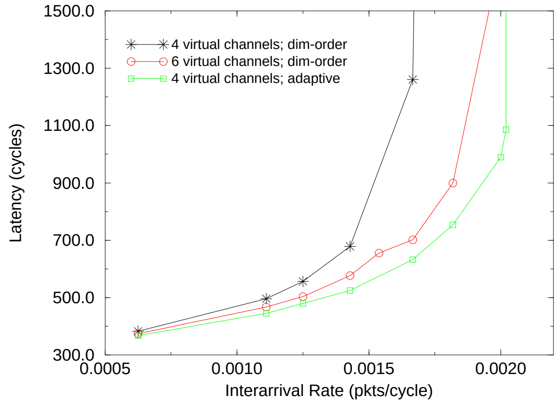


(a) Average latency for dimension-ordered routing



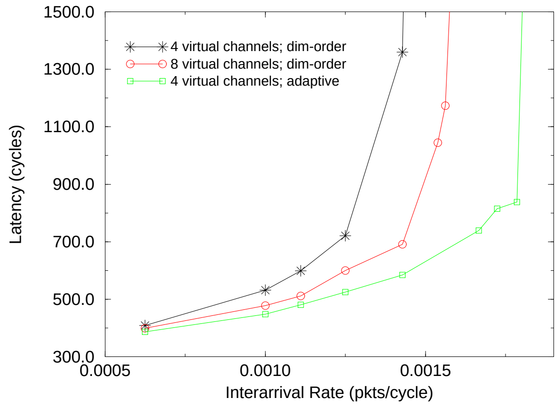
(b) Average latency for adaptive routing

Figure 3: Performance of wormhole algorithms under hot-spot traffic load.



Average latency for selected uniform traffic runs

Figure 4: Comparison of algorithms under uniform load.



Average latency for selected hot-spot traffic runs

Figure 5: Comparison of algorithms under hot-spot load.

5.1 Adding Virtual Channels

As Figure 2 shows, the performance of both algorithms under a uniform traffic load improves significantly as virtual channels are initially added. The benefits of added channels then diminishes as more are added. For dimension-ordered routing, a plateau is reached after 6 virtual channels. For the adaptive routing algorithm, this plateau is reached earlier at 4 virtual channels. Thus, a smaller number of virtual channels are needed for adaptive routing to reach the point where the benefits of additional channels is fairly diminished. Figure 3 shows a similar effect when the network is under a non-uniform load. From these results, it is clear that the performance gains provided by adding virtual channels depends on the routing algorithm employed. Adaptive algorithms tend to utilize virtual channels more effectively and thus, the addition of more channels does not have as large an impact as with static, dimension-ordered routing.

5.2 Comparison of Routing Algorithms

Using these simulations with the dimension-ordered and adaptive routing algorithms, we then compared the performance between the two algorithms as the number of virtual channels varied. We hoped to find a crossover point where dimension-ordered routing with additional virtual channels outperformed adaptive routing. This would indicate that the ability of virtual channels alone to reduce contention in the network was comparable to the ability of adaptive routing to do the same. Figure 4 shows the performance of dimension-ordered routing with 6 virtual channels and adaptive routing with 4 virtual channels under uniform load. The similarities in performance indicate that the additional virtual channels allow dimension-ordered routing to approach the performance of adaptive routing. When the effects of a faster implementation are taken into account, dimension-ordered routing may even outperform adaptive routing.

5.3 Varying the Traffic Pattern

Because adaptive routing is ideal for networks with non-uniform traffic patterns, we then did the same comparison using the results of the hot-spot traffic runs. Figure 5 shows the performance of dimension-ordered routing with 8 virtual channels and adaptive routing with 4 virtual channels under the non-uniform load. In contrast to the results with the uniform traffic, dimension-ordered routing does not approach the performance of adaptive routing even with the additional virtual channels. This indicates that when there are permanent hot links in the network, circumventing them is the only way to alleviate contention. Because these runs were done with a fixed hot-spot traffic pattern, it would be interesting to see how sensitive the performances of these algorithms are to traffic characteristics. For example, when the non-uniformities are transient, the performance of dimension-ordered routing with the additional channels may improve. With this in mind, for systems where the traffic pattern is known or can be predicted, an optimal router architecture can then be chosen.

6 Conclusion and Future Work

These experiments have shown how the performance of routing algorithms changes when the number of virtual channels and the traffic pattern varies for a particular router architecture.

There are many papers which have also looked at particular aspects of routing and switching in multicomputer networks, each assuming a particular architecture or set of parameters. As part of the work currently being done in this research group, a general evaluation framework is being developed to allow us to characterize the performance of multicomputer networks as a wide range of parameters are varied. Currently pp-mess-sim provides an environment where the routing algorithm, the switching scheme, the number of virtual channels, the allocation of virtual channels to subnetworks, the network size, the network topology, the destination distribution, the message generation distribution, the message length, and the types of evaluation metrics can be configured. In addition, specific to the cycle-level model of SPIDER, the speeds of each of the interfaces (network and host) and the sizes of its internal buffers can also be paramterized.

Currently, an abstract switch model is being developed which will allow us to additionally configure the architecture of the host (injection and reception ports) and the architecture of the switch (interconnection of channels, arbitration policies, queueing policies, channel multiplexing policies). From this model, we will be able to evaluate in a single framework, a variety of algorithms, network characteristics, and switch architectures.

Appendix A Classification of Related Work

Dally [9]:

Routing Algorithm: Dimension-ordered
Virtual Channels: 1,2,4,8,16
Virtual Channel Multiplexing: Random, deadline-based, age-based
Virtual Channel Allocation: Not specified
Network Size: Wide range
Topology: 2-ary n-fly (n=4,6,8,10), 2-ary n-cube (n=4,6,8,10,12,14), and 16-ary 2-cube.
Destination Distribution: Uniform
Message Generation Distribution: Poisson
Message Length: 20 flits
Switch Architecture: Source queueing modelled, Architecture not specified
Evaluation Metrics: Mean latency, latency distribution, throughput

Dally [13]:

Routing Algorithm: Non-minimal static-adaptive and dynamic-adaptive based on dimension reversal numbers, dimension-ordered routing. Selection function based on minimum congestion, maximum flexibility and straight line criteria.
Virtual Channels: 16
Virtual Channel Multiplexing: Not specified
Virtual Channel Allocation: Done according to threshhold
Network Size: 256
Topology: 16-ary 2-cube
Destination Distribution: Uniform and non-uniform
Message Generation Distribution: Random
Message Length: Not specified
Switch Architecture: Not specified, throttling of injection channels onto a subset of virtual channels.
Evaluation Metrics: Mean latency, throughput, throughput with faulty channels.

Duato [21]:

Routing Algorithm: Non-minimal fully adaptive algorithm based on Duato's methodology, dimension-ordered
Virtual Channels: 1 and 2
Virtual Channel Multiplexing: Time-dependent selection function
Virtual Channel Allocation: Not Applicable
Network Size: 64, 256, 1024, 4096
Topology: Binary n-cube
Destination Distribution: Uniform and exponential
Message Generation Distribution: Uniform across two parameters
Message Length: 16 and 256 flit messages
Switch Architecture: Crossbar interconnect, host modelled as an ideal sink (no contention), arbitration for output channels done with round robin.
Evaluation Metrics: Latency, throughput

Boppana [10]:

Routing Algorithm: 1) Dimension-ordered 2) Fully adaptive, minimal path: positive-hop, negative-hop, and negative-hop with wild cards 3) Partially adaptive, minimal path: North-last
Virtual Channels: Use as many as each routing algorithm requires
Virtual Channel Multiplexing: Pre-defined order depending on messages
Virtual Channel Allocation: Not applicable (minimum supplied)
Network Size: 256
Topology: 16-ary 2-cube
Destination Distribution: Uniform, hot-spot, local
Message Generation Distribution: Geometric
Message Length: 16 flits
Switch Architecture: Inject only if number of messages of same class is below a given threshhold.
Evaluation Metrics: Latency, throughput

Kim [11]:

Routing Algorithm: Dimension-ordered, planar-adaptive (minimal, partially-adaptive)
Virtual Channels: 1 or 2
Virtual Channel Multiplexing: Not specified
Virtual Channel Allocation: Not applicable
Network Size: 256 nodes
Topology: 2 dimensional mesh (square?)
Destination Distribution: Uniform
Message Generation Distribution: Not specified
Message Length: 24 flits, 256-1024 flits (hybrid traffic with 2 message lengths)
Switch Architecture: 2 injection channels and 2 sink channels: one for each message length.
Evaluation Metrics: Mean latency, latency distribution, standard deviation

Bibliography

[1] W. Athas and C. Seitz, "Multicomputers: Message-passing concurrent computers," *IEEE Computer*, pp. 9–24, August 1988.

[2] A. Tanenbaum, *Computer Networks*, Prentice-Hall, Englewood Cliffs, N.J., 1989.

[3] W. J. Dally and C. L. Seitz, "The torus routing chip," *Journal of Distributed Computing*, vol. 1, no. 3, pp. 187–196, 1986.

[4] S. Konstantinidou and L. Snyder, "Chaos router: Architecture and performance," in *Proc. Int'l Symposium on Computer Architecture*, pp. 212–221, May 1991.

[5] W. J. Dally et al., "Design and implementation of the message-driven processor," in *Proc. Brown/MIT Conference on Advanced Research in VLSI and Parallel Systems*, pp. 5–25, 1992.

[6] C. L. Seitz and W. Su, "A family of routing and communication chips based on the Mosaic," in *Symp. on Integrated Systems: Proc. of the Washington Conf.*, 1993.

[7] X. Zhang, "System effects of interprocessor communication latency in multicomputers," *IEEE Micro*, pp. 12–15,52–55, April 1991.

[8] W. J. Dally and C. L. Seitz, "Deadlock-free message routing in multiprocessor interconnection networks," *IEEE Trans. Computers*, vol. C-36, no. 5, pp. 547–553, May 1987.

[9] W. Dally, "Virtual-channel flow control," *IEEE Trans. Parallel and Distributed Systems*, vol. 3, no. 2, pp. 194–205, March 1992.

[10] R. Boppana and S. Chalasani, "A comparison of adaptive wormhole routing algorithms," in *Proc. Int'l Symposium on Computer Architecture*, pp. 351–360, 1993.

[11] J. H. Kim and A. A. Chien, "Evaluation of wormhole routed networks under hybrid traffic loads," in *Proc. Hawaii Int'l Conf. on System Sciences*, pp. 276–285, January 1993.

[12] S. Chittor and R. Enbody, "Performance evaluation of mesh-connected wormhole-routed networks for interprocessor communication in multicomputers," in *Supercomputing*, pp. 647–656, November 1990.

[13] W. Dally and H. Aoki, "Deadlock-free adaptive routing in multicomputer networks using virtual channels," *IEEE Trans. Parallel and Distributed Systems*, vol. 4, no. 4, pp. 466–475, April 1993.

[14] J. W. Dolter, S. W. Daniel, A. Mehra, J. L. Rexford, W. Feng, and K. G. Shin, "SPIDER: Flexible and efficient communication support for point-to-point distributed systems," Computer Science and Engineering Technical Report CSE-TR-180-93, University of Michigan, October 1993.

[15] W. Feng, J. Rexford, A. Mehra, S. Daniel, J. Dolter, and K. Shin, "Architectural support for managing communication in point-to-point distributed systems," Technical Report CSE-TR-197-94, University of Michigan, March 1994.

[16] D. Linder and J. Harden, "An adaptive and fault tolerant wormhole routing strategy for k-ary n-cubes," *IEEE Trans. Computers*, vol. 40, no. 1, pp. 2–12, January 1991.

[17] P. Berman, L. Gravano, G. Pifarre, and J. Sanz, "Adaptive deadlock- and livelock-free routing with all minimal paths in torus networks," in *Symposium on Parallel Algorithms and Architectures*, pp. 3–12, June 1992.

[18] C. Glass and L. Ni, "The turn model for adaptive routing," in *Proc. Int'l Symposium on Computer Architecture*, pp. 278–287, May 1992.

[19] A. A. Chien and J. H. Kim, "Planar-adaptive routing: Low-cost adaptive networks for multiprocessors," in *Proc. Int'l Symposium on Computer Architecture*, pp. 268–277, Australia, May 1992.

[20] J. Duato, "On the design of deadlock-free adaptive routing algorithms for multicomputers: Design methodologies," *Proc. Parallel Architectures and Languages, Europe*, pp. 390–405, June 1991.

[21] J. Duato, "Improving the efficiency of virtual channels with time-dependent selection functions," *Proc. Parallel Architectures and Languages, Europe*, pp. 635–650, 1992.

[22] J. Duato, "A new theory of deadlock-free adaptive routing in wormhole networks," *IEEE Trans. Parallel and Distributed Systems*, 1993.

[23] K. Bolding, "Non-uniformities introduced by virtual channel deadlock prevention," Technical Report UW-CSE-92-07-07, University of Washington, July 1992.

[24] J. Dolter, *A Programmable Routing Controller Supporting Multi-mode Routing and Switching in Distributed Real-Time Systems*, PhD thesis, University of Michigan, September 1993.

[25] K. G. Shin, "HARTS: A distributed real-time architecture," *IEEE Computer*, vol. 24, no. 5, pp. 25–35, May 1991.