Planar-Adaptive Routing: Low-Cost Adaptive Networks for Multiprocessors

ANDREW A. CHIEN AND JAE H. KIM

University of Illinois at Urbana-Champaign, Urbana, Illinois

Abstract. Network throughput can be increased by allowing multipath, adaptive routing. Adaptive routing allows more freedom in the paths taken by messages, spreading load over physical channels more evenly. The flexibility of adaptive routing introduces new possibilities of deadlock. Previous deadlock avoidance schemes in k-ary n-cubes require an exponential number of virtual channels [Linder and Harden, 1991]. We describe a family of deadlock-free routing algorithms, called *planar-adaptive routing* algorithms, that require only a constant number of virtual channels, independent of networks size and dimension. Planar-adaptive routing algorithms reduce the complexity of deadlock prevention by reducing the number of choices at each routing step. In the fault-free case, planar-adaptive networks are guaranteed to be deadlock-free. In the presence of network faults, the planar-adaptive router can be extended with misrouting to produce a working network which remains provably deadlock free and is provably livelock free. In addition, planar-adaptive networks can simultaneously support both in-order and adaptive, out-of-order packet delivery.

Planar-adaptive routing is of practical significance. It provides the simplest known support for deadlock-free adaptive routing in k-ary n-cubes of more than two dimensions (with $k > 2$). Restricting adaptivity reduces the hardware complexity, improving router speed or allowing additional performance-enhancing network features. The structure of planar-adaptive routers is amenable to efficient implementation.

Simulation studies show that planar-adaptive routers can increase the robustness of network throughput for nonuniform communication patterns. Planar-adaptive routers outperform deterministic routers with equal hardware resources. Further, adding virtual lanes to planar-adaptive routers increases this advantage. Comparisons with fully adaptive routers show that planar-adaptive routers, limited adaptive routers, can give superior performance. These results indicate the best way to allocate router resources to combine adaptivity and virtual lanes.

Planar-adaptive routers are a special case of limited adaptivity routers. We define a class of adaptive routers with f degrees of routing freedom. This class, termed *f-flat adaptive routers*, allows a direct cost-performance tradeoff between implementation cost (speed and silicon area) and routing freedom (channel utilization). For a network of a particular dimension, the cost of adaptivity grows linearly with the routing freedom. However, the rate of growth is a much larger constant for high-dimensional networks. All of the properties proven for planar-adaptive routers, such as deadlock and livelock freedom, also apply to f-flat adaptive routers.

Preliminary versions of some of this work appeared in *Proceedings of the 19th International Symposium on Computer Architecture*, May 1992, pp. 268–277, and *Proceedings of the 4th Symposium on Parallel and Distributed Processing*, December 1992.

Authors' address: Department of Computer Science, University of Illinois at Urbana-Champaign, 1304 West Springfield Avenue, Urbana, IL 61801.

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.
© 1995 ACM 0004-5411/95/0100-0091 $03.50

Categories and Subject Descriptors: B.4.3 [**Input / Output and Data Communications**]: Interconnections (Subsystems); C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design

General Terms: Algorithms, Performance, Reliability

Additional Key Words and Phrases: Adaptive routing, fault tolerance, interconnection networks, multicomputers, packet routing, parallel processing, transmission-order preservation

1. *Introduction*

In concurrent computers, interconnection networks are used by the processing nodes to exchange data and synchronize with each other. Network performance is often critical, as the performance of large-scale parallel machines is sensitive to network latency and throughput. While multicomputers have been touted as scalable parallel architectures, their scalability is limited by the performance of their interconnection networks.

An interconnection network is defined by its topology, routing, and flow control. The topology is the pattern of network node interconnection via physical communication channels. The routing algorithm specifies how packets choose paths through the network. Flow control deals with the allocation of channel and buffer resources to packets as they proceed through the network. This paper focuses on the issue of routing. In particular, we describe a class of adaptive routing algorithms which require only local information and modest hardware support. While our discussion is in the context of k-ary n-cube networks, the same ideas can be applied to other networks with regular topologies. Likewise, though we present the ideas in the context of *wormhole routing* [Dally 1987] the ideas apply to virtual cut through [Kermani and Kleinrock 1979] and store-and-forward networks as well.

2. *The Problem*

Most existing multicomputer routing networks use deterministic routing [Dally et al. 1989. Lillevik 1990; Seitz 1985; Seitz et al. 1988]. Although there are numerous paths between any source and destination, in order to avoid deadlock, *deterministic routing* defines a single path from source to destination. Fixed, single-path routing prevents effective use of the network's density of physical interconnection because the physical channels are allocated inflexibly. For any choice of fixed paths, some traffic patterns will produce performance which is much poorer than should be possible, given the physical interconnection. *Deterministic routing prevents the use of productive network resources to complete the communication.* An example of this situation is portrayed in Figure 1. In the figure, the circles represent nodes that are connected in a two-dimensional mesh. Four nodes are sending packets to the destinations indicated. In this case, the deterministic routing algorithm used to avoid deadlock requires routing horizontally, then vertically. Despite the presence of many other possible paths all four packets are forced to traverse the overloaded channel. The result is poor performance.

Adaptive routing allows packet routing to take advantage of the density of network interconnection. Typically, in adaptive schemes, packets can take one of a number of paths. This situation is depicted in Figure 2. In the figure, the same four packets are routed using an adaptive algorithm. Packet routes are

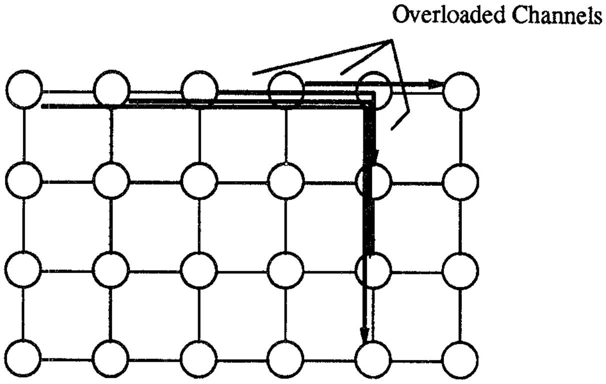


FIG. 1. Four packets and their routing paths under deterministic, dimension-order routing.

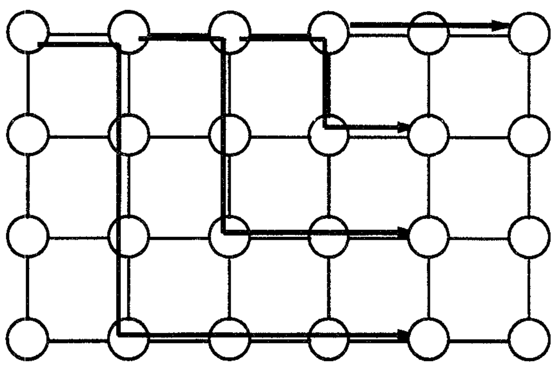


FIG. 2. The same four packets and potential routing paths under a more flexible routing scheme.

adapted based on whether channels are busy when a packet header arrives. If the desired physical channel is busy, another channel leading toward the destination may be chosen. In this case, the result is that none of the packets need to share physical channels. The key observation is that the best choice of paths depends on current network loading. To achieve good performance on all traffic loads, routing algorithms must be *adaptive*, for example, choose paths based on network conditions. The additional flexibility in adaptive routers can improve performance under both uniform and nonuniform loading. In general, in adaptively routed systems, path choices can be made on the basis of any local information. This is because the rapid fluctuations and traffic and intrinsic delay in obtaining remote information make it difficult to obtain accurate information. The distinction we make between deterministic and adaptive routing is sometimes termed oblivious versus non-oblivious routing [Borodin and Hopcroft 1985].

2.1. OVERVIEW. In the paper, we begin by briefly discussing some of the most relevant work in adaptive routers for k-ary n-cubes in Section 3. In particular, we highlight the cost of deadlock-avoidance in these systems. In Section 4, we introduce the *planar-adaptive* routing algorithm that allows adaptive routing at much lower hardware cost than previous techniques.

Planar-adaptive routing is proven to be deadlock-free in k-ary n-cubes with no wraparounds with only three virtual channels. Following that, we show how to extend the router to support fault-tolerance by adding disciplined misrouting. The resulting routing algorithm is termed a fault-tolerant, planar-adaptive router (PAR). Misrouting may result in nonminimal packet routes, but the routing algorithm remains deadlock-free. In addition, we prove that the fault-tolerant, planar-adaptive routing algorithm is livelock-free.

In Section 5, we consider the issue of message delivery order. The planar-adaptive routing algorithm can also simultaneously support both in-order and adaptive, out-of-order packet transmission. We demonstrate how this can be done. Section 6 discusses implementation issues for planar-adaptive routers with a focus on hardware complexity and speed. In Section 7, we evaluate a variety of PAR configurations, using several distinct traffic patterns. These results show that adaptive routing can improve performance, and the resources freed by using partially adaptive routing can be profitably applied in other ways. In Section 8, we show how planar-adaptive routing can be generalized to support f-degree adaptive routing in n-dimensional networks ($f \leq n$). The generalization defines a class of adaptive routers, *f-flat adaptive routers*, which provides deadlock-free routing algorithms for a full range of routing flexibility. Finally, Section 9 concludes the paper with a discussion of the ramifications of this work and some future directions.

3. *Background*

All adaptive routers choose from several paths based on channel loading, channel failure, or other dynamic information. The paths from which the router may select define a basic distinction among adaptive routers. This distinction separates adaptive routers which choose from a set of minimal paths (wasting no work) and nonminimal paths (potentially wasting routing work in exchange for increased routing freedom). These two types of adaptive routing algorithms are described below:

Minimal or productive adaptive routing. Packets are routed along paths of minimal distance to their destination. Packets never move away from their destination.

Nonminimal or misrouting-based adaptive routing. Packets may temporarily move away from their destination (misrouting), but eventually arrive at their destination. Due to misrouting, the distance a packet travels may not be minimal. The network may do some unnecessary work.

Significant work has explored nonminimal approaches [Blumenthal 1992; Konstantinidou and Snyder 1991; Ngai and Seitz 1989]. Although such efforts have produced interesting routers that can outperform minimal adaptive routers in some cases, the use of misrouting makes prevention of livelock a difficult problem. Complex schemes for packet priority and aging have been proposed, but they complicate the router logic required for routing and arbitration decisions. One interesting nondeterministic approach to livelock prevention is the Chaos router [Konstantinidou and Snyder 1991], but its efficiency and cost are still under evaluation. The use of misrouting in nonminimal adaptive routing also makes it difficult to preserve packet transmission order. Order-preserving packet delivery can significantly reduce the number of packets required to implement serialization protocols [Landin et al. 1991].

Minimal routers have a variety of attractive features. They do not waste any effort since packets are never transmitted away from their destination, so it is possible to utilize the full wire capacity[1] of the network productively. In addition, as we will show, minimal adaptive routers can support order-preserving routing efficiently. Finally, and perhaps most important, minimal routers allow message traffic to be confined to a subnetwork, providing advantages for both protection and performance predictability. For these reasons, we focus on minimal routers.

Planar-adaptive routing is a minimal, adaptive routing algorithm that is provably deadlock-free. Under appropriate reconfiguration algorithms, planar-adaptive networks are also fault-tolerant. In addition, planar-adaptive routing can be extended to support in-order packet delivery and adaptive, unordered packet delivery simultaneously. The designated in-order traffic is delivered in sequence, the other packets arrive in unspecified order. In all cases, the planar-adaptive networks are both deadlock and livelock free. The extension of planar-adaptive routing to higher degrees of routing freedom preserves these desirable properties.

3.1. THE COST OF DEADLOCK AVOIDANCE. Adaptive routing algorithms increase routing flexibility, multiplying the possibilities for deadlock. In multiprocessor networks, routing decisions must be made within a few cycles, and thus cannot incorporate complex distributed deadlock prevention and detection algorithms. Consequently, we focus on deadlock prevention schemes based on restricting the routing algorithm [Dally and Seitz 1989]. Because the physical network has cycles, naive routing schemes can produce deadlock, so we virtualize the network by introducing virtual channels (really only extra buffers) and define routing functions on the virtual networks. The new routing functions are from virtual channel to virtual channel and define a network with no cycles. This technique is useful in both deterministic and adaptive routing systems and is based on ideas from data networks found in Gelernter [1981] and Gunther [1981].

We focus particularly on two families of networks, k-ary n-cubes and multidimensional meshes (k-ary n-cubes without wraparounds) and assume wormhole routing [Dally 1987]. Dally and Seitz [1987] use virtual channels and restrict routing to implement dealock-free routing in k-ary and n-cubes. However, their scheme assumes a deterministic routing algorithm, dimension-order routing. A minimal adaptive routing scheme described by Linder and Harden [1991] extends the techniques of Dally and Seitz to allow adaptive routing while preventing deadlock. However, the hardware cost of their approach is significant, requiring a large number of virtual channels per physical channel. Such hardware complexity typically reduces attainable network speed.

Dally and Aoki [1993] have described an adaptive routing algorithm based on the notion of "dimension-reversals." Each dimension-reversal is a departure from dimension-order routing. In their static scheme, one additional virtual channel is required for each allowed dimension-reversal in order to prevent deadlock. Consequently, that scheme supports only a fixed number of dimen-

[1] We define *network capacity* as the bandwidth of all network wires being used simultaneously.

Topology	Determ.	Linder-Harden	Planar-Adaptive
2D Mesh	1	2	2
3D Mesh	1	4	3
4D Mesh	1	8	3
n-D Mesh	1	2^{n-1}	3

FIG. 3. Virtual channel requirements for deadlock-free routing algorithms for k-ary n-cubes. The virtual channel requirements for Dally and Aoki's scheme is omitted as they depend on the number of dimensions reversals permitted.

sion reversals. Their dynamic scheme can support an arbitrary number of dimension-reversals by tagging each message a dimension-reversal count and terminating adaptive routing if blocked by a packet with a lower number of reversals. Updating reversal counts and making routing decisions on that basis may increase router complexity significantly. Recently, Ni and Glass [1993] have described a deadlock-free adaptive router based on prohibiting turns that cause cycles. Direct comparison is difficult since descriptions of their approach leave open many design choices that may affect both performance and implementation complexity critically. For example, routing can be minimal or nonminimal.

Minimal adaptive routing algorithms developed by Duato [1991] and Berman [1992] require only a few virtual channels (2) per physical channel, but require that routing decisions be made based on accurate buffer status information. These routing algorithms are *dynamically* deadlock-free as the routing decisions ensure that no cycles can form. In contrast, the Linder–Harden [1991] approach, planar-adaptive routing, and the turn model are all statically deadlock-free, allowing flow control and routing to be decoupled. Statically deadlock-free routers can use arbitrary buffer sizes, selection policies, and simpler router decision logic. For the remainder of this paper, we focus on statically deadlock-free routers. We also do not consider adaptive schemes which are applicable only to binary hypercubes [Konstantinidou 1990] or those applicable only to meshes [Felperin et al. 1991] and not other types of k-ary n-cubes.

A table comparing the virtual channel hardware requirements of existing techniques and planar-adaptive routing is given in Figure 3. We include only those techniques that are deadlock-free over a variety of k-ary n-cubes, particularly those with larger radices ($k > 2$).

As can be seen in the figure, deterministic routing requires only one virtual channel to prevent deadlock for networks of all dimensions. All of the adaptive routing algorithms greater numbers of virtual channels. The Linder–Harden algorithm that permits fully adaptive routing requires a number of virtual channels exponential in n.[2] Practically speaking, the number is large even for a three- or four-dimensional network. In contrast, the planar-adaptive routing algorithm requires at most three virtual channels to prevent deadlock for networks of any dimension.

The cost advantage of planar-adaptive routing can be dramatic for higher-dimensional networks. Such networks are of increasing interest. Large-scale machines increase the effect of node latency, favoring networks with greater

[2]Any productive channel toward the destination may be used, and two 2^{n-1} virtual channels are required in the first dimension, 2^{n-2} in remaining dimensions.

connectivity. Looser constraints on wiring bisection and pin-bounds also make higher-dimensional networks more attractive. Furthermore, recent results indicate that higher-dimensional networks may exhibit more robust performance over a range of communication locality [Agarwal 1991].

3.1.1. *Cost Metrics.* In routing networks, the most expensive part is the wires for the physical channels. Following that, the second most expensive elements are the buffers and switching hardware (crossbar or other). Only if routing algorithms become very complex is routing logic a significant contributor. Since we will compare different routing schemes with respect to a fixed topology, the physical channel cost is constant. The issue is how to get the best performance out of the wires. Since the networks we consider are wormhole-routed, the main contribution to the buffering is the number of "virtual channels" which is the same as the minimum number of buffers. The number of switch ports in a cross bar affects both its cost and setup latency. Switch cost is superlinear in the number of ports, so partitionable switches will be cheaper and faster. We return to this issue when we discuss implementation issues in Section 6. Meanwhile, we focus on virtual channel requirements as a cost metric. Hardware cost savings are significant as they not only save chip area; they may also allow the router to run at higher speeds.

4. *Planar-Adaptive Routing*

Fully adaptive, minimal routing allows the use of any channels that move a packet toward its destination. As shown in Figure 4, packets are routed in the n-dimensional subcube of the network defined by the source and destination. It is possible to reduce the cost of deadlock avoidance by restricting routing flexibility (adaptivity). The motivation for this approach is the following observation about Linder and Harden's deadlock prevention scheme.

Observation. The requirement of a large number of virtual channels to prevent deadlock arises from the freedom to route in the n dimensions in arbitrary order.

By constraining the routing freedom to a few dimensions at a time, the hardware requirements for deadlock avoidance can be greatly reduced. Compared to fully adaptive routing, the planar-adaptive approach sacrifices some routing freedom to drastically reduce the possibilities of deadlock. Planar-adaptive routing limits routing freedom to two dimensions at a time. The reduced freedom makes it possible to prevent deadlock with only a fixed number of virtual channels, independent of the number of network dimensions.

Though there is less routing freedom than with fully adaptive routing, planar-adaptive routing still allows choice from a large number of paths from source to destination. Routing proceeds on a series of adaptive planes (two-dimensional surfaces). The routing freedom available in planar-adaptive routing is pictured in Figure 5. Within each adaptive plane, packets may use any channels leading toward their destination. Restricting the adaptivity in this fashion reduces the coupling between network dimensions, reducing the possibilities for interdimensional resource cycles. Consequently, fewer resources are needed to avoid deadlock.

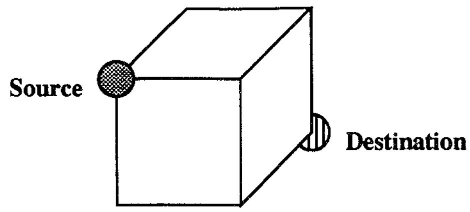


FIG. 4. A fully adaptive, minimal router allows a packet to be routed in the n-dimensional subcube defined by the source and destination. This subcube is shown for the case of a three-dimensional network.

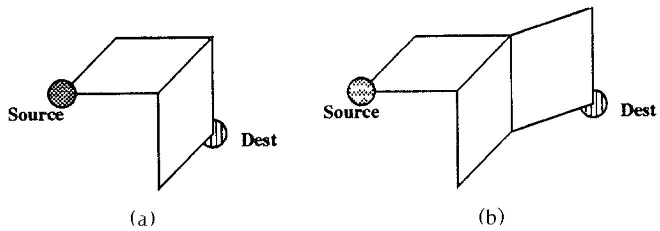


FIG. 5. Planar, adaptive routing allows the packet to be routes on a series of two-dimensional surfaces, adaptive planes. Planar-adaptive routing for three dimensions (a) and four dimensions (b) is illustrated. In the 4D case, the third adaptive plane is orthogonal to the first two.

4.1. NOTATION AND TERMINOLOGY. We adopt notation similar to that in Dally [1990] and Linder and Harden [1991]. We presume for the purposes of discussion, a physical topology of a k-ary n-cube with no wraparound paths.

$$k = \# \text{ of nodes along a dimension}$$
$$n = \# \text{ of dimensions, } 2^*n = \# \text{ of physical channels per node}$$
$$d_{i,j} = \text{dimension } i, \text{virtual channel } j$$
$$d_{i,j}+ = \text{dimension } i, \text{virtual channel } j, \text{ in the increasing direction}$$
$$d_{i,j}- = \text{dimension } i, \text{virtual channel } j, \text{ in the decreasing direction}$$

As a k-ary n-cube, the network has k^n nodes. $d_{i,j}$ specifies a set of virtual channels, j, in dimension i of the network. It specifies two for each node, one in the positive direction, one in the negative direction in the i dimension. We denote the positive direction channels as $d_{i,j}+$ and the negative direction channels as $d_{i,j}-$. Throughout this paper, we use the terms *message* and *packet* synonymously to indicate a unit of routing in the network.

4.2. PLANAR-ADAPTIVE ROUTING. The basic idea in planar-adaptive routing is to provide adaptivity in two dimensions at all times. Thus, a packet is routed adaptively in a series of two-dimensional planes. As the packet progresses toward its destination the routing dimensions change. Eventually, the packet is routed in all dimensions and delivered to its destination. By limiting the adaptivity to two dimensions and structuring the passage from one adaptive plane to the next, we reduce network cost while maintaining deadlock-freedom.

In a k-ary n-cube with no wraparound paths, planar-adaptive routing requires three virtual channels for each physical channel. We claim and will show that these three channels are sufficient to assure deadlock and livelock freedom.[3]

Define $n - 1$ adaptive planes, A_0 to A_{n-2}, as the combination of several sets of virtual channels.

$$A_i = d_{i,2} + d_{i+1,0} + d_{i+1,1}$$

Each adaptive plane involves only two dimensions.[4] Three virtual channels in each dimension are needed to support the $n - 1$ adaptive planes. More precisely, the first dimension needs only one virtual channel, but the maximum number of virtual channels for any dimension is three.

Planar-Adaptive Routing Algorithm:

High-level (between adaptive planes)

1. **For** $i = 0$, $i < (n - 1)$ **do**
   Route adaptively in A_i, see Low-level routing.
   end
2. After exiting the loop, it can only be necessary to correct the address in d_{n-1}. If necessary, route in $d_{n-1,2}$ to the destination.

Low-level (within adaptive plane A_i)

Adaptive plane A_i contains virtual channels $d_{i,2}$, $d_{i+1,0}$, and $d_{i+1,1}$. Within the plane, route adaptively with respect to dimensions d_i and d_{i+1} by choosing any channel that leads closer to the destination. In order to prevent deadlock, the traffic is divided into two classes: packets which need to increase (*increasing*) and decrease (*decreasing*) their d_i address. The virtual channels in A_i are divided into increasing and decreasing virtual networks which are completely disjoint (see Figure 6).

Increasing Network: $d_{i,2}+$, $d_{i+1,0}$
Decreasing Network: $d_{i,2}-$, $d_{i+1,1}$

1. Separate traffic into the appropriate subnetworks. Increasing traffic is routed in the increasing network. Decreasing traffic is routed in the decreasing network.
2. Within each, route packets adaptively toward their destination, using any of the productive channels.
3. When the d_i address is correct, routing is completed in plane A_i, so proceed to the next high-level step.

In high-level routing, the basic idea is to route successively in each adaptive plane. Routing in adaptive plane A_i reduces the distance in d_i to zero. After routing in all of the adaptive planes, the packet has reached its destination. For d_{n-1}, there cannot be any adaptivity left for a minimal router, so the packet is routed directly to its destination.

In low-level routing, the scheme is adaptive, as multiple paths can be chosen within each adaptive plane. In each adaptive plane, the packet completes its routing in at least one dimension. If in plane A_i, the d_{i+1} distance is reduced to zero first, routing continues in d_i exclusively, until the d_i distance is reduced to zero. The adaptive routing system is low cost as the overlapping adaptive

[3] It has been observed that this channel requirement can be reduced to two by substituting the $d_{i+1,0}$ channel in adaptive plane A_i for $d_{i+1,2}$ in adaptive phase A_{i+1}. Though it appears that this substitution preserves many of the properties of planar-adaptive routing, the change complicates the proofs significantly, making exact preservation unclear. In addition, the change may significantly complicate the routing algorithm as it requires the router to remember each message's current adaptive plane.

[4] The order of dimensions is arbitrary.

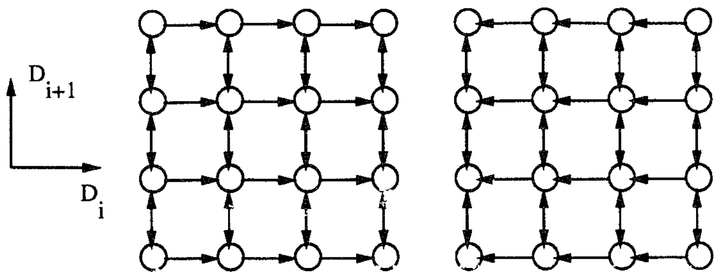


FIG. 6. The channels for an adaptive plane A_i. The increasing (a) and decreasing (b) networks are logically decoupled as they contain disjoint sets of virtual channels.

planes require only three virtual channels per physical channel for an n-dimensional network.

4.3. DEADLOCK FREEDOM. Planar-adaptive routing is deadlock-free. Our proof decomposes the network into the adaptive planes, shows that routing in each plane in deadlock-free, and then shows that cycles cannot form between planes. Consequently, the planar-adaptive routing algorithm is free from deadlocks.

THEOREM 4.3.1. *Planar-adaptive routing is deadlock free with only three virtual channels.*

LEMMA 4.3.2. *Routing within each adaptive plane, A_i, is deadlock free.*

PROOF OF LEMMA 4.3.2. Recall that each adaptive plane is divided into two networks which are completely separate.

Increasing Network: $d_{i,2}+$, $d_{i+1,0}$
Decreasing Network: $d_{i,2}-$, $d_{i+1,1}$

The packets in d_i+ can only depend on d_i+ and $d_{i+1,0}$ channels. The packets d_i- can only depend only on d_i- and $d_{i+1,1}$. Without loss of generality, we consider only the increasing network. Suppose a cycle forms within the increasing network. Because our grid is rectangular, the cycle must have a d_i- edge. Packets in the increasing network cannot use a d_i- edge, therefore no cycle can form. A symmetric argument applies for the decreasing network. □

LEMMA 4.3.3. *Routing between adaptive planes is deadlock-free. No cycles that cross between adaptive planes are possible.*

PROOF OF LEMMA 4.3.3. Each adaptive plane A_i routes traffic into A_{i+1}. Assume a cycle of size m occurs:

$$Cycle = A_{C1}, A_{C2}, \ldots, A_{Cm},$$

For each link in the cycle, the high-level algorithm assures that the adaptive planes are traversed in numerical order. This means that in our cycle, we must have $C_i < C(i+1)$. But the $<$ operator is transitive, this means by walking around the cycle, we can derive the constraint that $Ci < Ci$ which is a contradiction. Therefore, no cycles can form and routing between adaptive planes is deadlock-free. □

PROOF OF THEOREM. By Lemma 4.3.2, routing within each adaptive plane is deadlock-free. By Lemma 4.3.3, routing between adaptive planes is deadlock-free. There are no other dependences within the network; therefore, planar-adaptive routing is deadlock-free. □

Planar-adaptive routing allows some flexibility in routing at very low hardware cost. A primary concern in adaptive networks is the cost of deadlock prevention. To ensure deadlock-free routing, the planar-adaptive scheme requires only three virtual channels for k-ary n-cubes without wraparound channels. This is a dramatic improvement over previously published schemes [Dally and Aoki 1993; Ngai 1989]. As shown in Figure 3, the Linder–Harden deadlock prevention scheme requires 2^{n-1} virtual channels, a significant hardware overhead. Equally important, deadlock *and* livelock[5] are avoided without any complex reversal counting [Dally and Aoki 1993] or aging schemes [Ngai 1989]. This means that planar-adaptive routers will be very simple and can make routing decisions very rapidly. As a result, they may run faster than adaptive routers requiring complex techniques for deadlock and livelock prevention. While speed is important, planar-adaptive routers have other features which enhance their attractiveness. Planar-adaptive routers can be extended to support fault-tolerance and in-order message transmission.

4.4. FAULT TOLERANCE. Planar-adaptive networks can be augmented with misrouting to support fault-tolerance. The resulting networks can tolerate large numbers of faults, yet still deliver all packets to their destinations. The networks remain provably deadlock-free, and we give a proof of deadlock-freedom below. Circumventing faulty regions with only local information requires packet misrouting. Allowing misrouting also introduces the possibility of livelock, so we show that the resulting networks are livelock-free by proving that all packets make progress toward their destinations.

Planar-adaptive networks tolerate faults by routing around them. The basic idea is to use the flexibility of the adaptive routing algorithm to circumvent any faulty channels. We assume that all channel and node faults are detected. Then, all faulty regions are augmented until they are convex by running the deactivation algorithm given below. If faulty regions are not naturally convex, good nodes and channels are marked as faulty until the regions become convex. After the deactivation algorithm has been run, the convexity of faulty regions is ensured and packet routing can begin. Subject to the convexity constraint, planar-adaptive will route packets to all parts of the machine which remain connected. Our approach is the complement of that taken by Ngai [1989] where the working nodes form a convex region. Requiring the faulty regions to be convex allows a larger fraction of the nodes to remain in service for a given pattern of faults.

Fault-Tolerant, Planar-Adaptive Routing:

High-level. First, define a new adaptive plane with the previously unused channels below.

[5]Planar-adaptive routing without fault tolerance is trivially livelock free as only minimal routes are employed.

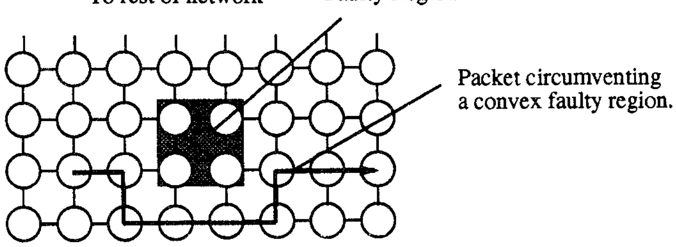


FIG. 7. A packet can be routed around convex faulty regions. A packet might get trapped by concave faulty regions.

$$A_{n-1} = d_{n-1,2} + d_{0,0} + d_{0,1}.$$

For adaptive planes A_0–A_{n-1}, route between adaptive planes using the low-level routing algorithm given below.

Low-level. For each adaptive plane, A_i, route the packet as follows (when $i + 1 = n$, use 0, i.e., compute $i + 1$ modulo n).

1. If not blocked by fault, route as in fault-free case.
2. If blocked by a fault in dimension d_{i+1}, route in d_i.
3. If blocked by a fault in d_i, route in d_{i+1}.
4. If blocked by a fault in d_i and the d_{i+1} distance has already been reduced to zero, it is necessary to misroute. If we were routing in d_{i+1}, continue to route in the same direction. If we were routing in d_i, the pick an arbitrary d_{i+1} direction and begin misrouting. At the first opportunity, route in d_i toward the destination, reverting to step 1.
5. Note that we cannot be blocked in d_{i+1} and have reduced the d_i distance to zero. If this were the case, we would have proceeded to the next adaptive plane.

Steps 2 and 3 make use of the adaptivity in routing to circumvent faults. Step 4 deals with the case where misrouting is necessary. Packets are routed around convex faulty regions in adaptive plane A_i by moving along d_{i+1} until reaching the corner of the faulty region. After reaching the corner, the packet follows the profile of the faulty region around until it has corrected its d_{i+1} address. An example is shown in Figure 7. Though step 4 may appear somewhat complicated, correct implementation is simple. Packets being misrouted are moving away from their destination. At first opportunity, such packets resume minimal routing toward their destination, resuming step 1.

4.4.1. *Node Deactivation Algorithm.* To ensure that the faulty regions are convex, despite arbitrary channel failures, nodes deactivate themselves and mark channels according to the following algorithm for two-dimensional networks:

Mark-Faulty-Nodes( )

1. Initially, mark all edge links as faulty.
2. Each channel or node detected as faulty is marked as such. A faulty node marks all of its channels as faulty.
3. Perform the **Convexify-Corner**( ) function for the four corner nodes, **Convexify**( ) function for the rest.
4. If some corner nodes were marked faulty, move those corner nodes in away from the edges, otherwise DONE.
   New corner nodes are chosen by moving the boundary with the most "faulty" corner nodes toward the center. Of course, both corner nodes that are on this edge should be moved.
5. **Goto** Step 3.

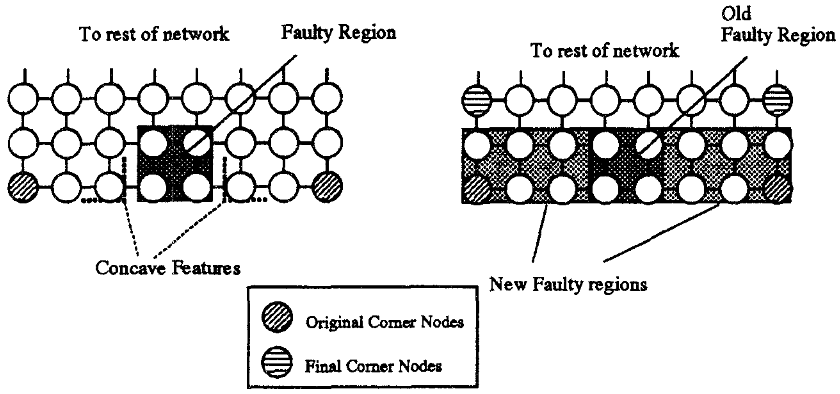


FIG. 8. A faulty region which abuts the boundary of a k-ary n-cube with no wraparounds forms a concave feature. The additional nodes indicates will be deactivated by our convexity-ensuring algorithm.

Convexify( ). *If a node has two or more channels in different dimensions marked faulty, then it should mark itself and all of its channels as faulty.*
Convexify-Corner( ). *If a node has more than 2 channels marked faulty, then it should mark itself as faulty. The states of its channels are not changed.*

The high-level algorithm, **Mark-Faulty-Nodes**( ) gradually moves the corner nodes in until a straight boundary in each dimension can be straight. This makes the external boundary of the working region of nodes convex (no intrusions of faulty nodes). Figure 8 illustrates initial and final corner nodes.

The **Convexify**( ) algorithm makes all internal faulty regions convex. Informally, it ensures convexity because any concave feature in a faulty region must have a boundary node (just outside it) that has at least two channels in different dimensions that are faulty. This boundary node sits at the concave feature. By our algorithm, no such node can exist, as it would have added itself to the faulty region, removing the concave feature.

However, there is a problem at the boundary with this simple deactivation algorithm. It marks all the nodes as faulty. The reason for this is that the corner nodes will see two faulty channels (n for higher-dimensional networks) at the edges and deactivate themselves. If we did not distinguish the corners, they would move toward the center until all nodes have been marked faulty.

We prevent this occurrence by using a different marking rule for the corner nodes, **Convexify-Corner**( ). This means that the hardware must have some mechanism for designating nodes as corners. From the actual hardware corners of the machine, the algorithm iterates, moving the corners inward and eventually finding corners that allow the external boundary of the region of working nodes to be convex.

Some convex and nonconvex faulty regions are shown in Figure 9. The ideas behind our two-dimensional algorithm can be extended to higher-dimensional networks by marking boundary edges to handle higher-dimensional networks. Our marking scheme will reduce the number of nodes available, however, this may be a reasonable compromise for a minimal increase in router complexity to support fault tolerance. Devising a routing scheme that maintains maximal connectivity and deadlock-freedom requires nonlocal information and perhaps

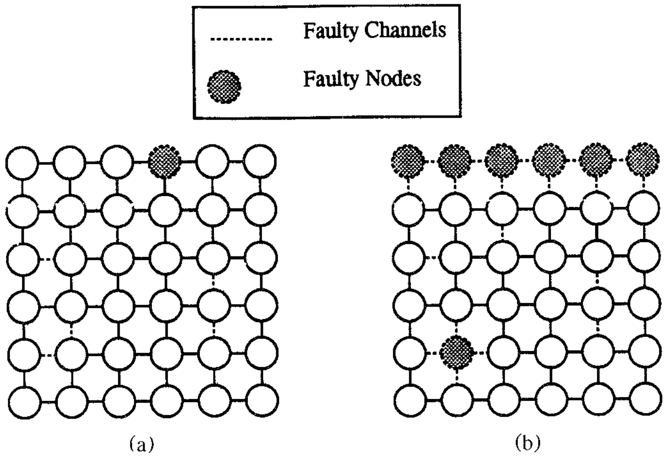


FIG. 9. The deactivation removes any concave features and makes them convex. The picture illustrates the cases of a single faulty channel, a pair of faulty channels in the same dimension, a pair of faulty channels in different dimensions, and a faulty node on the boundary. (a) Before convexity enforcement. (b) After faulty regions have been made convex.

programmable routing tables. The incorporation of such features is likely to significantly reduce router speed.

4.4.2. *Misrouting and Deadlock Freedom.*

THEOREM 4.4.1. *Fault-tolerant, planar-adaptive routing is deadlock-free.*

LEMMA 4.4.2. *Fault tolerant, planar-adaptive routing is deadlock-free within an adaptive plane.*

PROOF OF LEMMA 4.4.2. Consider adaptive plane A_i. Misrouting only occurs when a message is forced to "overshoot" its destination in d_{i+1}. Overshoot in dimension d_i does not occur if the d_i distance ever reaches zero because then the packet would pass into plane A_{i+1}. Consider the two adaptive networks in the plane:

Increasing: $d_{i,2}+$, $d_{i+1,0}$
Decreasing: $d_{i,2}-$, $d_{i+1,1}$

Without loss of generality, assume we are in the increasing network and we overshoot increasing d_{i+1}. Overshooting in d_{i+1} adds no additional dependences between virtual channels, as routing continues in the $d_{i,2}+$, $d_{i+1,0}+$. When we reach the end of the d_{i+1} edge of the faulty region, we resume routing in d_i+ toward the destination. Upon reaching the other side of the faulty region, we resume minimal routing toward the destination. All of this routing occurs in the increasing network (virtual channels $d_{i,2}+$, $d_{i+1,0}$). If the d_i distance is reduced to zero before circumventing the faulty region, we go to the next plane, where routing continues normally. No new dependences have been added that did not exist in the network with no faults. Therefore, no deadlocks can have been introduced. By symmetry, this argument applies to both the increasing and decreasing networks. Thus, fault-tolerant, planar-adaptive routing is deadlock free within an adaptive plane. □

PROOF OF THEOREM 4.4.1. By Lemma 4.4.2, fault-tolerant, planar-adaptive routing is deadlock-free within each adaptive plane. As in the fault-free case, packets only pass from lower-numbered adaptive planes to higher ones. Therefore, fault-tolerant, planar-adaptive routing is deadlock-free. □

THEOREM 4.4.3. *Fault-tolerant, planar-adaptive routing will route packets to all nonfaulty nodes in the presence of convex faulty regions.*

LEMMA 4.4.4. *In the presence of convex faulty regions, within each adaptive plane, A_i, planar-adaptive routing eventually reduces the distance to the destination in d_i to zero. This ensures that packets eventually make progress to the next adaptive plane.*

PROOF OF LEMMA 4.4.4. If there are no faults, the proof follows from the monotonic reduction in d_i or d_{i+1} distance with each routing step. Eventually, the d_i distance is reduced to zero.

If there are faults, misrouting may cause the d_{i+1} distance to increase temporarily. However, the d_i distance is monotonic nonincreasing. This can be seen from the routing algorithm within an adaptive plane. For each misrouting to circumvent a faulty region, the d_i distance decreases by at least one. Each misrouting requires at most $k - 1$ steps (the maximum distance in one dimension). We cannot hit the boundary, as such would be a concave feature (see Figure 8). Faulty regions are convex, so they must have a corner. When the corner is reached, turning and routing in d_i reduces the d_i distance by at least one. Thus, the d_i distance must be reduced by at least one for each k routing steps. A packet can only experience k misroutings in one plane, so that maximum number routing steps in one adaptive plane is k^2. The lemma is proved. □

PROOF OF THEOREM 4.4.3. Assume there exists a nonfaulty node p that the fault-tolerant, planar-adaptive routing cannot reach. That means at some stage we are blocked from making progress toward the destination. By Lemma 4.4.4 routing in each adaptive plane A_i must complete successfully within k^2 routing steps. Successful completion implies that the distance in dimension d_i is reduced to zero, and it never again increases. The one exception to this rule is d_0, which we reuse in the final adaptive plane A_{n-1}. This case is considered below. Thus, at the end of $(n - 1) * k^2$ steps, are guaranteed to have routed successfully in all dimensions except d_{n-1}.

In the final adaptive plane (A_{n-1}), by Lemma 4.4.4, routing will successfully reduce the d_{n-1} distance to zero. However, if after doing so, we cannot reach p, it must be that we are blocked by faults in d_0. When we entered A_{n-1}, the d_0 distance was zero. Because our faulty regions are convex, we know that any errors introduced in d_0 are immediately correctable on the other side of the faulty region. Thus, the only time that we route successfully in d_{n-1}, yet cannot correct d_0 is if the destination node lies within a faulty region. This means that node p must be faulty. This contradicts our assumption. Therefore, fault-tolerant planar adaptive routing will route packets to all nonfaulty nodes in the presence of convex faulty regions. □

4.4.2. *Livelock Freedom.* Supporting fault tolerance requires the introduction of misrouting. If packets can move away from their destinations, there is a possibility of livelock.

THEOREM 4.4.5. *Planar-adaptive routing is livelock-free.*

PROOF OF THEOREM 4.4.5. Each packet makes progress toward its destination. The maximum length of any misrouting is k, the extent of the network in any one dimension. For each k misrouting steps in d_{i+1}, the packet makes at least one step toward its destination in dimension in d_i. Since all packets are guaranteed to make progress toward their destination, there can be no livelock. □

As the scale and application domain of parallel machines continue to increase, fault-tolerance is of growing importance. Planar-adaptive routing supports the implementation of fault-tolerance with several desirable characteristics. First, there is virtually no overhead when systems are operating fault-free. Second, the routing scheme in the presence of faults remains simple, so good performance is possible. Finally, the reconfiguration algorithm for nodes and their channels is simple and uses simple, local operations to ensure that all faulty regions in the network are convex.

5. *Order-Preserving Packet Transmission*

For some classes of traffic, preserving packet transmission order is important. Landin and Haridi showed that in-order packet delivery, "race-free" networks, can significantly reduce the number of packets required to implement serialization protocols for coherent shared memory [Landin et al. 1991]. Not only does the lesser packet requirement reduce network loading, it also may improve a critical performance feature, memory operation latency, by eliminating waits for acknowledgments in the protocol. In a number of other shared-memory protocols, preserving message-transmission order may also simplify the protocol and/or improve its performance [Chaiken et al. 1990; Dubois et al. 1988].

Our minimal, adaptive routing algorithms support both in-order packet delivery and adaptive, unordered packet delivery simultaneously. All traffic must be specified as ordered or unordered. Planar-adaptive routing guarantees in-order packet delivery for the ordered traffic both in the fault-free case, and in the presence of static faults.[6] For simplicity, we assume only one "virtual lane" [Dally 1992] per virtual channel, so packets routed along a single path cannot pass each other. In a multilane virtual channel system, passing can be prevented by limiting the ordered traffic for each source-destination pair to a single lane.

In any network that uses multipath routing, preserving packet transmission order is difficult. Selecting paths dynamically, as in adaptive routing systems, only worsens the situation. Data networks such as the Arpanet use packet sequence numbers and reordering (message reassembly). Such schemes are impractical in multiprocessor routing networks as maintaining sequence numbers for hundreds or thousands of nodes, both incoming and outgoing, is

[6] The set of faulty channels is assumed not to be changing during the packet routing.

expensive. Further, the reassembly control and buffer space impose a significant hardware overhead.

Order Preserving, Planar-adaptive routing:
This algorithm applies for networks with no faulty channels. Route as in planar-adaptive routing algorithm, with the following change to the low-level algorithm:
1. In adaptive plane A_i, preferentially route in d_i. No steps in d_{i+1} will be taken in A_i.

This change reduces the planar-adaptive scheme to dimension-order routing. It is more restrictive than the planar-adaptive algorithm, so it is guaranteed to be deadlock free, even in the presence of unordered, adaptive traffic. Informally, since dimension-order routing uses only one path between each source and destination, packets traveling along the same path cannot pass each other, so transmission order is preserved. Actually, the algorithm is overly restrictive; any consistent choice for all packets heading for a particular destination is sufficient. For example, one way to spread traffic more evenly is to use bits from a different dimension in the address (such as that for dimension $i + 3$ in adaptive plane A_i) to bias routing in the adaptive plane, ordering the messages with different d_{i+3} destination addresses on different paths. Biasing the adaptive choices has no affect on the deadlock-freedom of the routing algorithm.

Order Preserving, Fault-tolerant, Planar-adaptive routing:
This algorithm applies to networks with a static set of faulty channels, guaranteed to be in convex regions as described above.

High-level. Route as in Fault-tolerant, planar-adaptive routing, with the following modified low-level routing algorithm. The only change is shown in bold. Note also that in step 4, return to step 1 now refers to our modified step 1.

Low-level. For each adaptive plane, A_i, route the packet as follows.

Route as in the fault-tolerant, planar-adaptive algorithm with the following change to steps 1 and 4, the other steps remain the same. The choice of which particular d_{i+1} direction does not matter, so long as it is the same for all messages between a particular source-destination pair.

1. If not blocked by fault, **route in** d_i.
4. If blocked by a fault in d_i and the d_{i+1} distance has already been reduced to zero, it is necessary to misroute, If we were routing in d_{i+1}, continue to route in the same direction. **If we were routing in d_i, then pick a particular d_{i+1} direction and begin misrouting**. At the first opportunity, route in d_i toward the destination. Continue to route in d_i only until it is possible to correct d_{i+1}. At that point, route in d_{i+1} to distance zero in this dimension[7], then revert to step 1.

Since we have only modified the algorithm to bias our "adaptive" choices, the network is still deadlock-free and guaranteed to deliver all messages. It does not matter what method is used to choose the d_{i+1} misrouting direction so long as it is deterministic for all packets between a source and destination pair.

THEOREM 5.1. *The order-preserving, fault-tolerant, planar adaptive routing algorithm preserves message transmission order.* (*In the proof, we refer to it as simply the "routing algorithm"*).

PROOF OF THEOREM 5.1. The routing algorithm specifies a unique path through the network for each source-destination pair. At each step of the routing algorithm, there is no choice in which channel the packet will be routed

[7]This must be possible, as faulty regions are convex.

in next. Since there is a single path, and packets on the same path may not pass each other, in-order packet delivery is assured. □

The planar-adaptive routing scheme can be generalized to support both in-order and out-of-order packet delivery traffic simultaneously. Under such conditions, deadlock and livelock freedom is still assured. The motivation for this is simple. In many cases, implicit synchronization may render the preservation of packet transmission order unnecessary. Such traffic should profit from the full advantage of adaptive routing. In other cases, as we have discussed, the preservation of transmission order makes an important difference in algorithm and communication efficiency. A particularly important case is implementing memory coherence protocols.

6. *Implementation Issues*

Planar-adaptive routing is of practical significance. Restricting the adaptivity reduces the hardware cost of adaptive routing by reducing the number of virtual channels required to prevent deadlock. Fewer virtual channels may allow the construction of faster routers (fewer buffer loads to drive), or the additional hardware available may be used to incorporate other performance enhancing features such as more "virtual lanes" [Dally 1992].

Planar-adaptive routing is relatively low cost. Compared to Linder and Harden's scheme, planar-adaptive routing requires significantly fewer virtual channels. Planar adaptive routing requires only three buffers per physical channel, allowing the construction of very small, fast routers. In addition, the overall hardware resources consumed by the router will be smaller, broadening the applicability of the routers.

Planar-adaptive routers allow significantly lower switching complexity. The planar-adaptive algorithm allows an internal router organization without large crossbars. Even without considering the overhead for deadlock-prevention, a fully adaptive router must be able to connect each input to $n + 1$ outputs. This is necessary to allow routing in arbitrary dimensions with equal speed. Supporting the symmetric adaptivity with high performance usually implies a number of $n \times n$ crossbars—one for each virtual network (see Figure 10). As shown in Figure 11, the switching hardware in a planar-adaptive routing can be partitioned into smaller crossbars of size 4×4. This significantly reduces the amount of hardware required and should reduce the time to set up and drive data across the switches. Low connectivity requirements also make it possible to use organizations which allow the router performance to be further optimized for high-speed, low-latency performance [Dally et al. 1989; Dally and Song 1987].

In planar-adaptive routers, the routing function prevents deadlock, completely independent of the flow control. *No routing decisions depend on the presence or absence of flits in particular network buffers*. This allows routing and flow-control decisions to be made separately, decoupling the hardware control structures and potentially increasing performance. Of course, buffer occupancy information may be used to bias routing choices. But, such use differs in that it is purely optional and need not affect critical timing paths.

The extensions to planar-adaptive routing for fault tolerance and packet ordering add only slightly to the router complexity. Fault tolerance requires only slightly more complex routing logic. Message that require in-order trans-

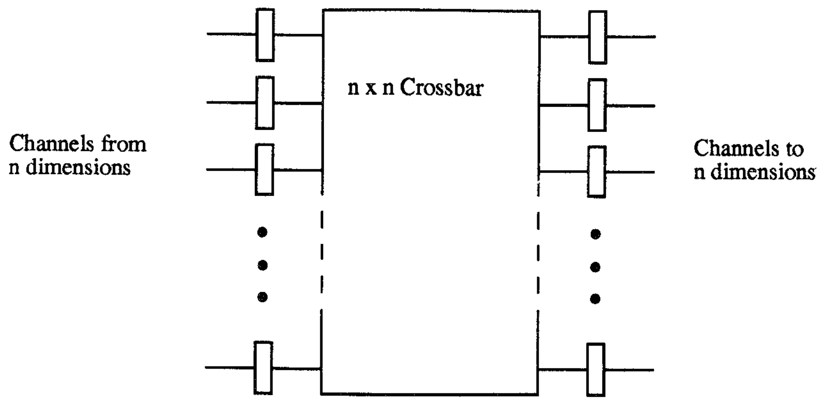


FIG. 10. The minimal connectivity requirements for a fully adaptive router in a k-ary n-cube requires at least $n \times n$ connectivity. This is shown here as a crossbar.

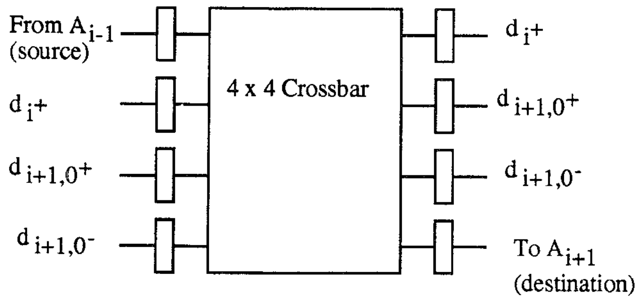


FIG. 11. One crossbar of the two required for a planar-adaptive router module, A_i. The crossbar is only 4×4, permitting high-speed connections. A planar-adaptive router can be built by composing such modules.

mission are tagged with a single bit. This tag serves as an input to the routing logic, biasing the adaptive choices to a preferred dimension and reducing the routing to single path, thereby producing order-preserving routing.

One pragmatic concern in adaptive routing techniques concerns the encoding of destination information into packet headers. For packet routers to have single-cycle node latency, they must receive all routing inputs as soon as the message begins to arrive. In addition, they must make routing decisions in a few cycles. In fully adaptive routers, the routing inputs may include the entire destination address, not just a few dimensions. Fitting the entire address into a single-cycle channel transfer (many networks have one or two byte-wide channels) is difficult. Planar-adaptive routing limits the routing inputs to only two dimensions at a time. Fitting two dimensions of information into a single cycle channel transfer is much easier. As a packet is routed in successive adaptive planes, different information comes to the front of the message header as routing information becomes unnecessary and is stripped off.

7. *Performance*

We explore the performance of planar-adaptive routers, comparing their performance to both dimension-order and fully adaptive routers. Evaluation of a network's performance can be achieved through analytical modeling or simulation. Unfortunately, in networks supporting wormhole routing, nodes cannot be analyzed separately since their queues have strong interactions and coupled event transitions. Moreover, adaptive routers have complex behavior which depends on network status. Due to the difficulty of accurate analytic modeling, router performance is evaluated via simulation.

For all simulations, the packet size is 24 flits, *flow-control units*. Each flit can be transferred across a channel in a single cycle. If flits are 8 bits each, this short packet size is comparable to a cache line or a procedure invocation record. All of the loads are normalized with respect to the network's maximum achievable performance, which is determined by the network's bisection bandwidth. Our networks have two separate unidirectional channels between adjacent nodes.

Our studies focus on answering one question: How does the performance of planar-adaptive routing compare to deterministic and fully adaptive routing with similar resources? First we compare planar-adaptive routing to dimension-order routing under a variety of traffic patterns. Second, we compare planar-adaptive routers to fully adaptive routers, compensating for the greater virtual channel requirements of fully adaptive routers by using planar-adaptive routers with virtual lanes. Our simulations show, not only can planar-adaptive routers outperform the deterministic routers, they can also outperform fully adaptive routers. With equal resources, the planar-adaptive router provides superior performance to both the fully adaptive router and the deterministic router.

7.1. TRAFFIC PATTERNS. Throughout the performance evaluation, we use three different traffic loads which are described below.

Random (*uniform*). Each node sends with equal probability to all other nodes in the system.

Dimension-reversal. Each node sends messages to a node with address of reversed dimension index. In two-dimensional networks, node (x, y) communicates with node (y, x). This gives the same traffic as a matrix transpose. For three-dimensional networks, we use an analogous traffic pattern, in which node (x, y, z) sends messages to node $(y, x, \sqrt[3]{N} - z)$. In four-dimensional networks, node (x, y, z, w) communicates with node (y, x, w, z).

Bit-reversal. A node with address $abcd_2$ sends messages to a node $dcba_2$.

The Random traffic load has been used to evaluate both deterministic and adaptive routers in many previous simulation studies and provides an important point of reference. The other two nonuniform loads, Dimension-reversal and Bit-reversal, have been used to demonstrate the effectiveness of adaptive routing to dissipate congestion, as they both create significant congestion under dimension-order routing. Both nonuniform loads are extended to three- and four-dimensional traffic patterns that induce similar kinds of congestion. Although these traffic patterns are by no means exhaustive, they represent some interesting extrema of the space of possible patterns. Because of the relative immaturity of software on parallel machines, characteristics traces of communication patterns are not available.

A planar-adaptive router can be characterized by the number of virtual lanes in each adaptive plane. Each plane consists of three virtual channels: one major channel ($d_{i,2}$ along dimension i in plane A_i) and two minor channels ($d_{i+1,0}$ and $d_{i+1,1}$ along dimension $i + 1$ in plane A_i). We denote capacity of the planar-adaptive router with the number of virtual lanes in each of these three classes, (major, minor, minor). For example, virtual channel numbers $(2, 1, 1)$ means two virtual lanes for the major virtual channel, and a single lane for each minor virtual channel.

7.2. PLANAR-ADAPTIVE ROUTING VS. DIMENSION-ORDER ROUTING. While allowing more freedom in path selection, the flexibility of adaptive routing introduces new possibilities of deadlock. Routing flexibility implies an increased cost in hardware resources. To evaluate the merits of an adaptive routing algorithm, we must measure the performance gain and weigh that against the cost of increased hardware resources. In this section, we evaluate the benefits of adaptive routing by comparing the performance of a PAR and a dimension-order router with equal resources. We allot equal numbers of virtual channels to both routers. Since the dimension-order router does not need any virtual channels for deadlock prevention, all of the extra channels are used as virtual lanes. For the PAR, extra channels not required for deadlock prevention are also used as virtual lanes. Router configurations simulated are summarized in Figure 12(a) and (b). A PAR for a two-dimensional network requires only one virtual channel for the x-dimension and two virtual channels for the y-dimension. Thus, a $(2, 1, 1)$ PAR for a two-dimensional network has the same number of virtual channels as a dimension-order router with two virtual channels per physical channel.

As before, we compare the performance of both routers on two-, three-, and four-dimensional mesh networks. The effect of virtual lanes on network performance depends on traffic patterns. To observe the dependence, we repeat the comparative study under previous traffic patterns: uniform random, dimension-reversal, and bit-reversal. For each network and traffic pattern, two kinds of router configurations were simulated:

1. With the minimum number of additional lanes, we match the number of virtual channels for the dimension order and planar-adaptive routers.
2. Configurations with approximately equal numbers of virtual channels, but twice the number in a minimum configuration.

7.2.1. *Uniform Traffic.* Figure 13 compares the performance of PARs and dimension-order routers under uniform random traffic. Dashed lines show the results for a minimum number of virtual channels, and solid lines show the results for a doubled number of virtual channels. For all of the networks studied, two-, three-, or four-dimensional mesh, PARs gave slightly worse performance then dimension-order routers under uniform traffic. Not surprisingly, the figure shows that PARs deliver messages a little faster, but saturate the networks earlier than dimension-order routers.

However, the figure shows a noticeable result: As the number of virtual lanes increases, the performance of PARs catches up to that of dimension-order routers even under uniform traffic loads. In two-dimensional networks, as shown in Figure 13(a), the PAR $(2, 1, 1)$ is outperformed by the dimension-order

Dimension	Dimension-order router		PAR	
	Configuration	Total VC per node	Configuration	Total VC per node
2	2 vl/link	8	(2,1,1)	8
3	2 vl/link	12	(1,1,1)	12
4	2 vl/link	16	(1,1,1)	18

(a)

Dimension	Dimension-order router		PAR	
	Configuration	Total VC per node	Configuration	Total VC per node
2	4 vl/link	16	(4,2,2)	16
3	4 vl/link	24	(2,2,2)	24
4	4 vl/link	32	(2,2,2)	36

(b)

FIG. 12. Virtual channel allocations. (a) Equalized with the minimum number of virtual channels. (b) Equalized with double the minimum number of virtual channels.

router with two virtual lanes. However, the PAR $(4, 2, 2)$ shows a nearly identical performance to that of the dimension-order router with an equal quantity of resources. In higher-dimensional networks, the catch-up effect is even more pronounced. In three- and four- dimensional networks, PARs $(2, 2, 2)$ are even better than dimension-order routers, giving identical peak throughput and lower latency.

Previous published results [Dally 1992; Kim and Chien, 1992] show that virtual lanes can increase network throughput, but the incremental benefit of each virtual lane decreases with each additional lane. For the given quantity of resources, PARs partition them statically between deadlock prevention and virtual lanes. Thus, compared to the dimension-order router, the PAR has relatively fewer virtual lanes for an equal number of virtual channels. Consequently, the incremental benefit of adding virtual lanes to the PAR is larger, allowing the PAR's performance to catch up to that of the dimension-order router.

7.2.2. *Nonuniform Traffic.* Under nonuniform traffic loads, dimension-order routers provide poor performance. As under uniform traffic, it is an interesting question to ask if adding virtual lanes improves network performance under nonuniform traffic loads.

Figure 14 shows the simulation results under dimension-reversal traffic loads. With an equal quantity of resources, PARs give a much better performance than dimension-order routers. Even under nonuniform traffic, the addition of virtual lanes significantly improves the performance of PARs. However, adding virtual lanes to dimension-order routing does not improve and can even reduce network performance under the dimension-reversal traffic loads. Consequently, the performance difference between PARs and dimension order routers with virtual lanes is significant.

Figure 15 shows the results of simulations under bit-reversal traffic loads. As with dimension-reversal traffic, adding virtual lanes does not improve the performance of dimension-order routers at all. On the other hand, it signifi-

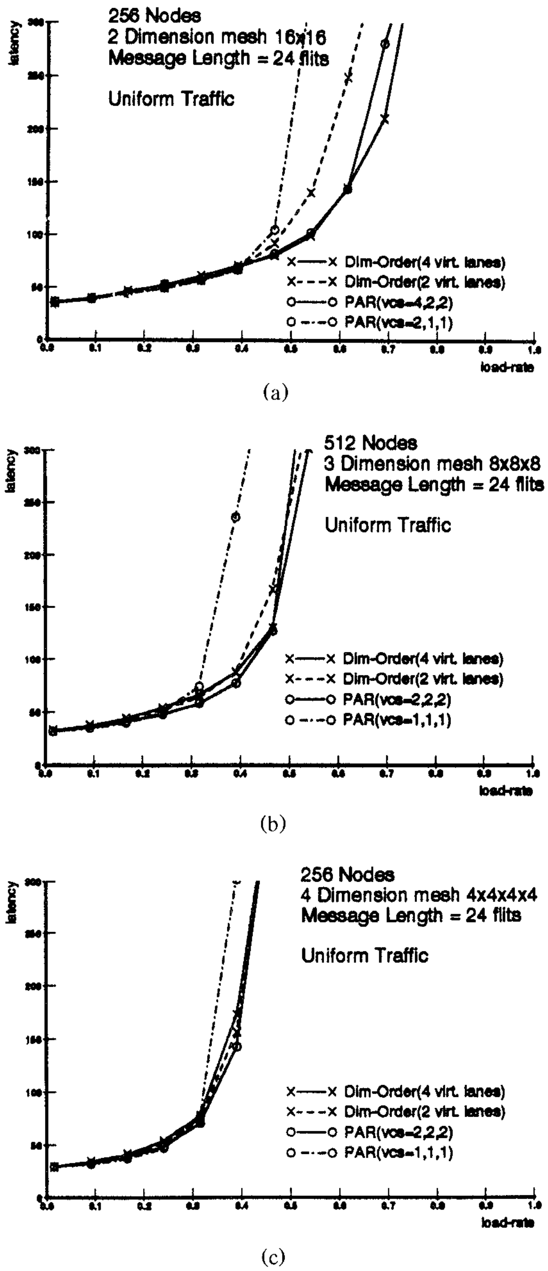
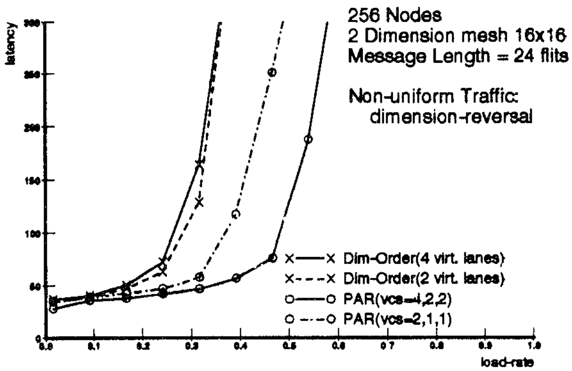


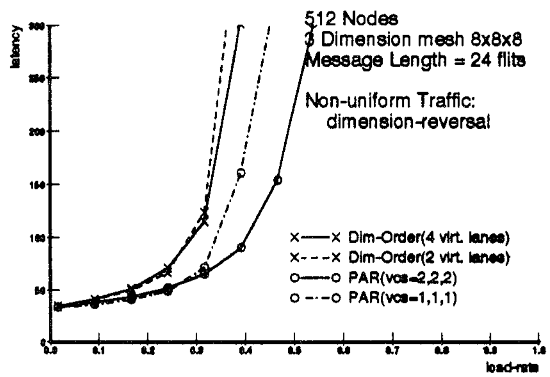
FIG. 13. Latency of dimension-order routing and planar-adaptive routing with uniform traffic on (a) 16×16 2D, (b) $8 \times 8 \times 8$ 3D, and (c) $4 \times 4 \times 4 \times 4$ 4D mesh networks.

cantly improves the PAR's performance, magnifying the performance difference between PARs and dimension-order routers.

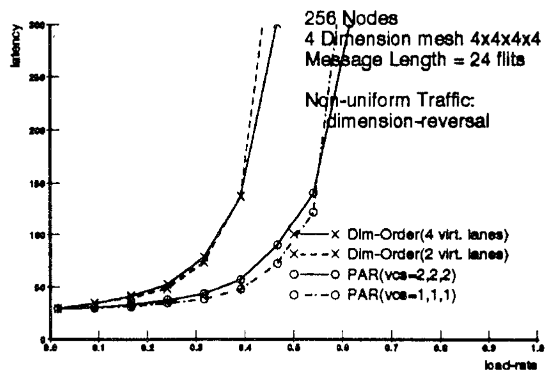
The comparative study of the effects of virtual lanes shows several facts: (1) Virtual lanes increase the peak throughput of planar-adaptive routing for all of the traffic loads considered. (2) In contrast, virtual lanes improve performance of dimension-order routing only under uniform traffic loads; there is no benefit for nonuniform traffic. (3) Thus, under nonuniform traffic loads, the PAR significantly outperforms the dimension-order router with equal hardware



(a)



(b)



(c)

FIG. 14. Latency of dimension-order routing and planar-adaptive routing with dimension-reversal traffic on (a) 15 × 16 2D, (b) 8 × 8 × 8 3D, and (c) 4 × 4 × 4 × 4 4D mesh networks.

resources. (4) Even under uniform traffic loads, the PAR provides identical performance to the dimension-order router.

7.3. PLANAR-ADAPTIVE ROUTING VS. FULLY ADAPTIVE ROUTING. Having shown that PARs can outperform dimension-order routers, we compare PARs with fully adaptive routers to answer the question "How much adaptivity

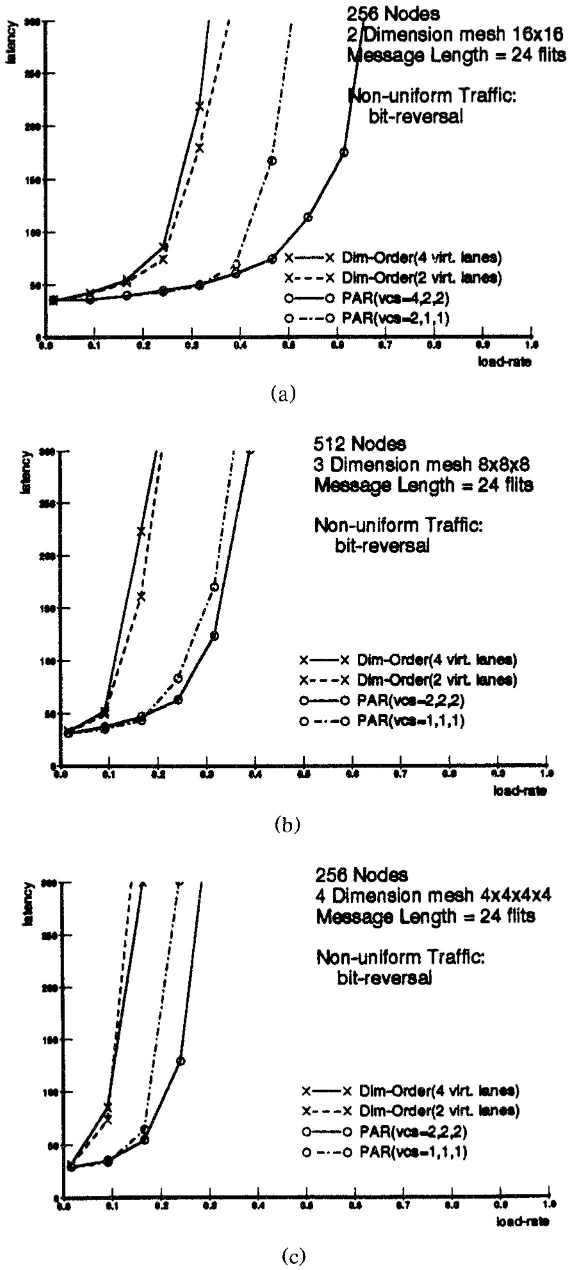


FIG. 15. Latency of dimension-order routing and planar-adaptive routing with bit-reversal traffic on (a) 16×16 2D, (b) $8 \times 8 \times 8$ 3D, and (c) $4 \times 4 \times 4 \times 4$ 4D mesh networks.

produces optimal performance?" Planar-adaptive routing reduces the hardware resources required for deadlock prevention, so to give a fair comparison we augment the PAR with virtual lanes until it has a comparable number of virtual channel resources. In all cases, the fully adaptive router is compared with a PAR with equal or fewer hardware resources.

The fully adaptive router is Linder and Harden's fully adaptive routing algorithm which allows all possible minimal paths [Linder and Harden 1991]. In n-dimensional mesh networks, their model requires 2^{n-1} virtual channels in

Dimension	PAR		Fully adaptive Router	
	Configuration	Total VC	Configuration	Total VC
3	(2,2,2)	24	4 vc/link	24
4	(2,2,2)	36	8 vc/link	64

FIG 16. Virtual channel allocations for comparing planar-adaptive and fully adaptive routers.

the last dimension, 2^{n-2} in all other dimensions. To model a symmetric router design, we simply allot 2^{n-1} of virtual channels to all dimensions. The additional channels other are simply used as virtual lanes.

The router configurations studied are summarized in Figure 16. For two-dimensional networks, a planar-adaptive router is fully adaptive, so our experiments consider only three and four dimensional networks. For three-dimensional networks, equal numbers of virtual channels are provided to each router, but for four-dimensional networks, the fully adaptive router requires a large number of virtual channels (64) for deadlock prevention, so instead, we compare to a PAR with only half as many resources.

7.3.1. *Uniform Traffic.* The PAR outperforms the fully adaptive router under uniform traffic in both three and four dimensional networks (see Figure 17). In three-dimensional networks, the planar-adaptive router shows lower latency over a wide range of load rates and much higher throughput. In four-dimensional networks, with only half as many resources, the PAR outperforms the fully adaptive router, providing slightly lower latencies and higher peak throughput. The PAR's superior performance arises from the uneven channel utilization of mesh networks; for all minimal adaptive routers, most alternative paths in a mesh lead through the center, worsening the load imbalance and causing saturation at lower loads. In fully adaptive routers, routing freedom in all dimensions causes high congestion at the center of the network. Planar-adaptive routing reduces this congestion by confining traffic to two-dimensional planes, keeping the utilization of channels between planes balanced. This shows that under uniform traffic loads, lower adaptivity is preferable as it preserves the even distribution of traffic over the network. Too much adaptivity may degrade network performance under uniform traffic on k-ary n-cube networks.

7.3.2. *Nonuniform Traffic.* The simulation results for nonuniform traffic loads show that the performance of both adaptive routing algorithms depends on the specific traffic pattern (see Figures 18 and 19). Under dimension-reversal traffic, the fully adaptive router's performance is much worse than that of the PAR (see Figure 18). In the three-dimensional network, the peak throughput under PAR is much higher. In the four-dimensional network, while the throughput of PAR is only a little higher, remember the PAR has only one half the resources of the fully adaptive router. As with uniform traffic, the poor performance of the fully adaptive router under dimension-reversal traffic is due to uneven channel utilization caused by too much adaptivity.

Under bit-reversal traffic, the fully adaptive router outperforms the planar-adaptive router (see Figure 19). The relatively poor performance of the PAR arises from the unique traffic distribution of bit-reversal traffic. In the $8 \times 8 \times$

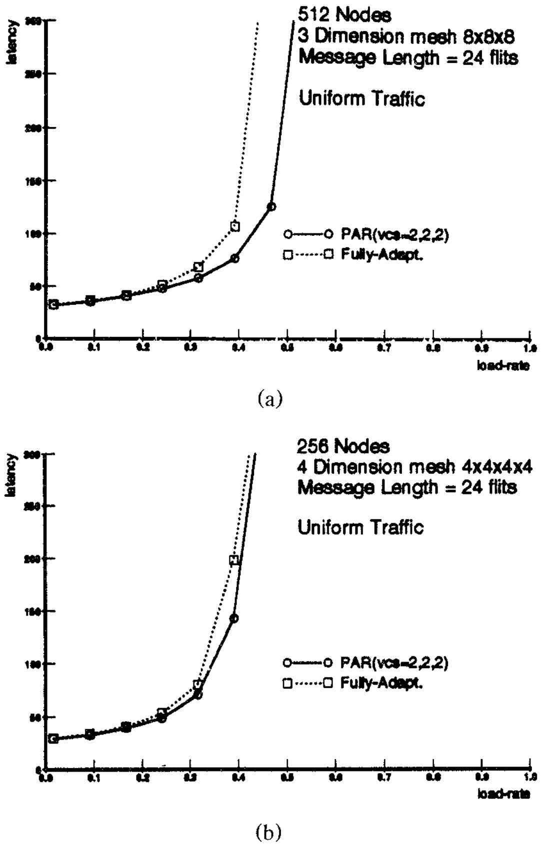
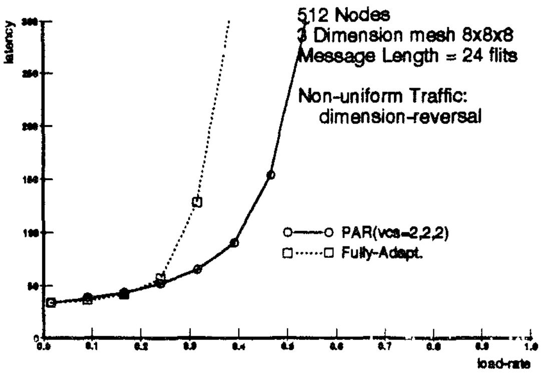


FIG. 17. Comparison of planar-adaptive routing and fully adaptive routing under uniform random traffic. (a) $8 \times 8 \times 8$ 3D and (b) $4 \times 4 \times 4 \times 4$ 4D mesh networks. The 4D PAR has many fewer virtual channels than the 4D fully adaptive router.

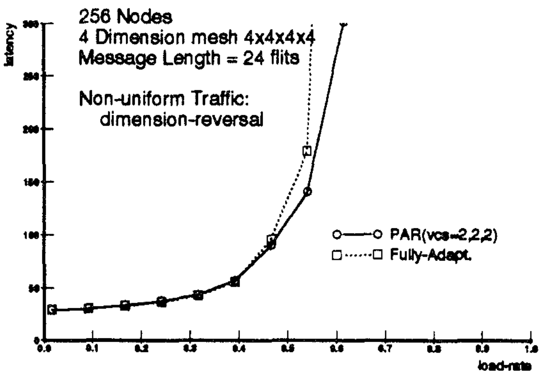
8 three-dimensional network, half of the total traffic is confined to four XZ-planes. For such messages, planar-adaptive routing is reduced to dimension-order routing since there is no adaptivity in the y-dimension. Although changing the order of dimensions would eliminate the effect for this traffic pattern, the same poor performance would occur for a different permutation. This is a traffic pattern where performance suffers from limited adaptivity. Despite that fact, performance is only slightly worse for PARs.

In summary, we have shown that planar-adaptive routing is more economical for high-performance networks by comparing the performances of the PAR and the fully adaptive router. Under uniform traffic, the PAR showed much better performance than the fully adaptive router with equal numbers of virtual channels. Under nonuniform traffic loads, the PAR provided comparable performance to the fully adaptive router, much better performance under dimension-reversal but slightly poorer performance under bit-reversal traffic.

7.4. DISCUSSION. Our experiments show that the performance of both adaptive routing schemes depends critically on the traffic patterns. Under uniform traffic, the PAR outperformed the fully adaptive router. Under dimension-reversal traffic, the PAR significantly outperformed the fully adaptive router, but under bit-reversal traffic, PAR gave poorer performance. Thus, a more comprehensive conclusion requires experimental with a wider range of



(a)



(b)

FIG. 18. Comparison of planar-adaptive routing and fully adaptive routing under dimension-reversal traffic. (a) $8 \times 8 \times 8$ 3D and (b) $4 \times 4 \times 4 \times 4$ 4D mesh networks. The 4D PAR has many fewer virtual channels then the 4D fully adaptive router.

traffic patterns, or perhaps using traffic loads taken from existing multicomputer systems.

8. *Higher Degrees of Adaptivity*

Planar-adaptive routing represents a restricted form of adaptive routing, allowing only two degrees of routing freedom. We have focused on planar-adaptive routing to this point, because it implies the least hardware complexity and because it is not clear *how much* adaptivity is beneficial. In this section, we consider a generalized family of f-flat adaptive routers (f degrees of routing freedom) for k-ary n-cubes. First, we outline their cost and describe how the properties proven for PARs can be extended to f-flat adaptive routers. We use the term *f-flat* adaptive router, because an f-flat is an f-dimensional subspace of an n-dimensional space.

Planar-adaptive routing can be extended to f-flat adaptive routing by generalizing the notion of an adaptive plane to an adaptive f-dimensional space. Planar-adaptive routers allow routing in two dimensions at a time. In adaptive plane A_i, routing is allowed in dimensions i and $i + 1$. In and f-flat adaptive router, a packet passes through a series of adaptive *flats*, F_i, each an f-dimensional space. While PARs choose from only two possibilities, two degrees of routing freedom, f-flat adaptive routers choose from f choices, providing f degrees of routing freedom.

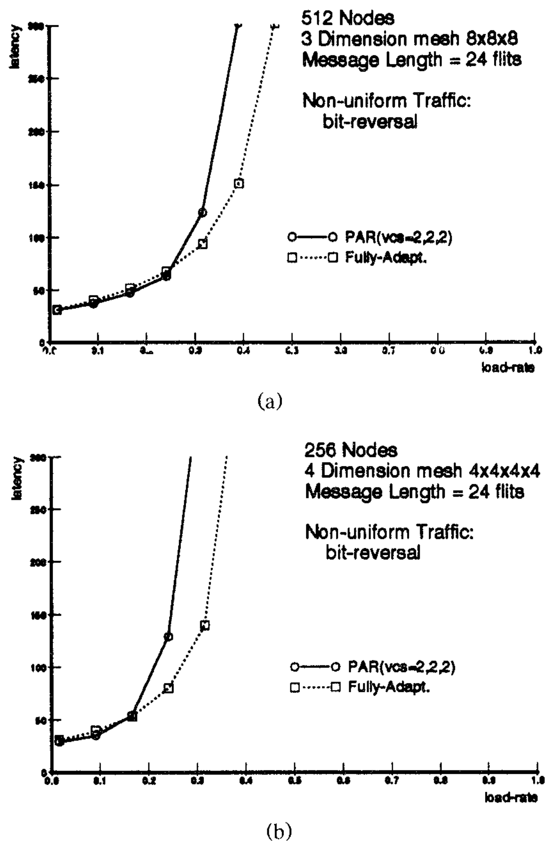


FIG. 19. Comparison of planar-adaptive routing and fully adaptive routing under bit-reversal traffic. (a) $8 \times 8 \times 8$ 3D and (b) $4 \times 4 \times 4 \times 4$ 4D mesh networks. The 4D PAR has many fewer virtual channels than the 4D fully adaptive router.

Using f-flat routers allows the designer to increase routing freedom independent of the network dimension. For example, it is possible to use a 2-flat (planar), 3-flat, 4-flat, or even 5-flat adaptive router in a k-ary 5-cube. The 5-flat adaptive router is fully adaptive. However, the increased routing freedom is not without cost. Increasing f will dramatically increase the router complexity and hardware resources required to prevent deadlock. If the f-flats are composed as in the planar-adaptive router (route dimension i to completion in F_i then pass to F_{i+1}), then there is substantial overlap in dimensions between successive f-flats.[8] Using the Linder–Harden algorithm for deadlock-free routing within each f-flat, the virtual channel requirements are approximately the number of virtual channels per f-flat times the number of overlapping f-flats.

More specifically, for a k-ary n-cubes without wraparounds the virtual channel requirements can be written as follows:

$$VC(mesh) = 2^{f-1} + 2^{f-2} * \min(f-1, n-f).$$

This is the maximum number of virtual channels used in any dimension and the last $f-1$ dimension are routed in the last f-flat. Where the two terms arise from the 2^{n-1} virtual channels for the first dimensions and the 2^{n-2} virtual channels in each successive dimension required by the Linder–Harden

[8]The f-flats could also be composed with decreasing degrees of overlap ranging from $f-1$ dimensions, as we have described, all the way to 0. However, reducing overlap is a less effective way of reducing router cost by decreasing routing freedom. Reducing f is much more effective.

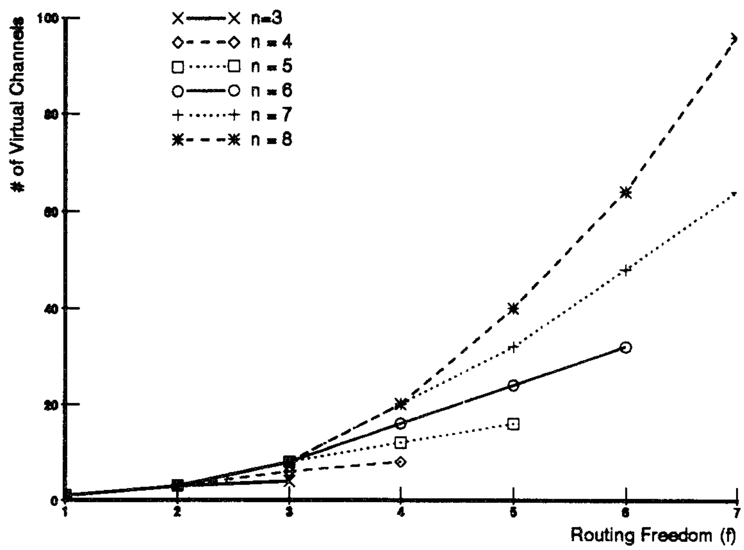


FIG. 20. The cost for deadlock-free routing for various degrees of routing freedom in k-ary n-cubes with no wraparounds. Surprisingly, each additional degree of routing freedom costs a linear increase in resources.

scheme. The *min* function arises from the number of overlapping f-flats which depends on the relative size of f and $n/2$. If f is smaller, the overlap is $f - 1$. If f is larger, the number of overlapping f-flats is determined by $n - f$. The number of virtual channels required for a variety of f-flat adaptive routers is presented in Figure 20.

Although the number of virtual channels required appears to increase exponentially with routing freedom, for a network of given dimension, the increase is effectively linear. This is due to a compensating decrease in the amount of overlap between successive adaptive f-spaces as f approaches n. However, as is clearly shown in Figure 20, the cost of deadlock-free routing increases much more rapidly for higher-dimensioned networks. At the two extremes, no-adaptivity and full adaptivity, the cost of the f-flat routers matches that of dimension order routing and the Linder–Harden fully adaptive scheme, respectively. In terms of practicality, only those routers with moderate virtual channel requirements can be implemented without significant speed penalty.

Increasing the adaptivity has a direct impact on the crossbar switch sizes required. Larger crossbars may reduce the achievable router speed, so the benefits of adaptivity must be weighed against their implementation cost. The precise size of crossbar required depends on how the crossbars can be partitioned, but in general at least an $(f + 1) \times (f + 1)$ crossbar will be required for an f-flat adaptive router.

Extending the properties proven for PARs is straightforward. For each property, we briefly describe how to extend the proofs.

—*Deadlock Freedom.* Routing within each f-flat is deadlock-free based on Linder and Harden [1991]. Cycles cannot form between f-flats as packets always proceed through the f-flats in increasing order. Therefore routing is free from deadlocks.

—*Fault Tolerance.* In general, the planar-adaptive faulty channel and node-marking algorithm works fine for f-flat routing freedom. The f-flat adaptive router has a variety of choices for fault-tolerance, since misrouting can be done in any of $f - 1$ dimensions. Any dimension can be used to circumvent the *convex* faulty regions. This only increases the possibilities for circumventing a faulty region, so fault-tolerance remains possible. Exploiting the increased opportunity for fault-tolerance may require larger crossbars. For example, the Linder–Harden scheme partitions the crossbars so that only the last dimension could be used for circumvention. If faulty regions are asymmetric in various dimensions, choosing a misrouting dimension adaptively or even randomly may improve performance.

—*Livelock Freedom.* The fault-tolerant f-flat adaptive router is also livelock-free, based on essentially the same proof as for the planar-adaptive case. Each message will make steady progress toward the destination in the major (lowest) dimension of routing in each adaptive f-flat. Therefore, f-flat adaptive routers are livelock-free.

—*Order Preservation.* As with planar adaptive routing, restricting traffic to single-path can be used to preserve message transmission order. F degrees of routing freedom provide additional choices provided which can be used to spread the ordered traffic over a larger number of virtual channels.

The family of f-flat adaptive routers makes it possible for network designers to choose an appropriate level of adaptivity in routing. The degree of routing freedom can be chosen on the basis of effectiveness in increasing channel utilization (network throughput) and the cost in terms of hardware complexity and network clock speed. Thus, with families of adaptive routers that give the choice over a range of adaptivity, network designers can make cost-performance trade-offs.

9. *Conclusions*

We have presented a simple class of adaptive routers for k-ary n-cubes, called *planar-adaptive* routers. These routers are provably deadlock-free and simple enough for high-performance implementation. Specifically, their virtual channel requirements are fixed and do not grow as the dimension of the network is increased. Planar-adaptive routing can be used to build adaptive networks for meshes of arbitrary dimension with only three virtual channels. This is less than the exponential number of virtual channels required by the Linder–Harden scheme even for three-dimensional networks. If higher-dimensional networks are considered, planar adaptive routers are *much* simpler. In addition, restricting the adaptivity in routing also allows the router switches to be partitioned. Not only does this reduce their cost, but it also makes it possible to tune them for low latency and high speed.

We have also described several simple extensions to the basic planar adaptive router. These extensions support fault-tolerant and order-preserving packet transmission. Both extensions require minimal hardware support, no additional virtual channels, and are provably deadlock and livelock-free. If two degrees of routing freedom are not sufficient, planar-adaptive routing can be generalized to produce a class of routers with a range of routing freedom. These f-flat adaptive routers are deadlock free and retain the desirable properties of planar adaptive routers (save simplicity). F-flat adaptive routers quantify the cost of routing freedom, allowing network designers to make a direct cost-performance trade-off.

We have also evaluated the effectiveness of planar-adaptive routing using simulation on a variety of traffic loads. Our studies show that planar-adaptive routers provide more robust performance than deterministic routers. Perhaps more significant, our simulations show that adaptivity and virtual lanes are complementary techniques. Virtual lanes increase peak throughput of planar-adaptive routing for *all* the traffic loads considered. These results suggest that the best way to spend network resources is on the combination of adaptivity and virtual lanes. Because of their limited adaptivity, planar-adaptive routers require less resources for deadlock prevention. Using *adaptivity* and *virtual lanes* together allows planar-adaptive routers with less hardware resources to deliver comparable or even superior performance to fully adaptive routers.

We are currently pursuing construction of hardware prototypes to evaluate the cost of adaptive routers. Based on these designs, we are pursuing a careful characterization of the cost of a variety of router extensions with great interest [Aoyama and Chien 1994; Chien 1993].

Fundamental to the evaluation of limited adaptivity routers lies a deeper question. How much adaptivity do routing networks need? This question will only be answered as application programs and software systems for massively parallel machine mature.

An unanswered question is how to best make use of the flexibility and control offered by limited-adaptivity routers. One current problem in mesh-based multicomputers is the sharing of communication resources among multiprogrammed jobs. The sharing such resources can lead to unpredictable performance. The ideas in planar-adaptive routing provide a way of allowing flexible use of network resources, yet restricting their use.

REFERENCES

AGARWAL, A. 1991. Limits on interconnection network performance. *IEEE Trans. Paral. Dist. Syst.* 2, (4), 398–412.

AOYAMA, K., AND CHIEN, A. A. 1994. The cost of adaptivity and virtual lanes in a wormhole router. *J. VLSI Design*, Special Issue on Interconnection Networks.

BERMAN, P., GRAVANO, L., PIFARRE, G., AND SANZ, J. 1992. Adaptive deadlock- and livelock-free routing with all minimal paths in torus networks. In *Proceedings of the 4th Annual ACM Symposium on Parallel Algorithms and Architectures* (San Diego, Calif., June 29–July 1). ACM, New York, pp. 3–12.

BLUMENTHAL, D., CHEN, K., MA, J., FEUERSTEIN, R., AND SAUER, J. 1992. Demonstration of a deflection routing 2 × 2 photonic switch for computer interconnects. *IEEE Photo. Tech. Lett. 4*, 1, 169–73.

BORODIN, A., AND HOPCROFT, J. 1985. Routing, merging and sorting on parallel models of computation. *J. Comput. Syst. Sci. 30*, 130–45.

CHAIKEN, D., FIELDS, C., KURIHARA, K., AND AGARWAL, A. 1990. Directory-based cache coherence in large-scale multiprocessors. *IEEE Computer*, June.

CHIEN, A. A. 1993. A cost and performance model for k-ary n-cube wormhole routers. In *Proceedings of Hot Interconnects Workshop* (Aug).

CHIEN, A. A., AND KIM, J. H. 1992. Planar-adaptive routing: Low-cost adaptive network for multiprocessors. In *Proceedings of the 19th Annual International Symposium on Computer Architecture* (Gold Const, Australia, May 19–21). ACM, New York, pp. 268–277.

DALLY, W. J. 1987. *A VLSI Architecture for Concurrent Data Structures*. Kluwer Academic Publishers, Boston, Mass.

DALLY, W. J. 1990. Performance analysis of k-ary n-cube interconnection networks. *IEEE Trans. Comput. 39*, 6 (June).

DALLY, W. J. 1992. Virtual channel flow control. *IEEE Trans. Paral. Dist. Syst. 3*, 2, 194–205.

DALLY, W. J., AND AOKI, H. 1993. Deadlock-free adaptive routing in multicomputer networks using virtual channels. *IEEE Trans. Paral. Dist. Syst 4*, 4 (Apr), 466–474.

DALLY, W. J., CHIEN, A., FISKE, S., HORWAT, W., KEEN, J., LARIVEE, M., LETHIN, R., NUTH, P., WILLS, S., CARRICK, P., AND FYLER, G. 1989. The J-Machine: A fine-grain concurrent computer. In *Information Processing 89, Proceedings of the IFIP Congress*. pp. 1147–1153.

DALLY, W. J., AND SEITZ, C. 1987. Deadlock-free message routing in multiprocessor interconnection networks. *IEEE Trans. Comput. C-36*, 5.

DALLY, W. J., AND SONG, P. 1987. Design of a self-timed vlsi multicomputer communication controller. In *Proceedings of the International Conference on Computer Design*. IEEE Computer Society Press, New York, pp. 230–234.

DUATO, J. 1991. On the design of deadlock-free adaptive routing algorithms for multicomputers: Design methodologies. In *Proceedings of Parallel Architectures and Languages Europe*. Springer-Verlag, Berlin.

DUBOIS, M., SCHEURICH, C., AND BRIGGS, F. 1988. Synchronization, coherence and event ordering in multiprocessors. *IEEE Comput.* (9-21).

FELPERIN, S., GRAVANO, L., PIFARRÉ, G., AND SANZ, J. 1991. Fully-adaptive routing: Packet-switching performance and wormhole algorithms. In *Proceedings of Supercomputing '91* (Albuquerque, N. M., Nov. 18–22). ACM, New York, pp. 654–663.

GELERNTER, D. 1981. A dag-based algorithm for prevention of store-and-forward deadlock in packet networks. *IEEE Trans. Comput. C-30*, 10:12 (Oct.) pp. 709–715.

GLASS, C., AND NI, L. 1992. The turn model for adaptive routing. In *Proceedings of the 19th Annual International Symposium on Computer Architecture*, (Gold Coast, Australia, May 19–21). ACM, New York, pp. 278–287.

GUNTHER, K. D. 1981. Prevention of deadlocks in packet-switched data transport systems. *IEEE Trans. Commun. COM-29*, 4.

KERMANI, P. AND KLEINROCK, L. 1974. Virtual cut-through: A new computer communications switching technique. *Computer Networks* 3, 4, 267–86.

KIM, J. H., AND CHIEN, A. A. 1992. An evaluation of planar-adaptive routing (PAR). In *Proceedings of the 4th Symposium on Parallel and Distributed Processing*. IEEE Computer Society Press, New York, pp. 470–478.

KONSTANTINIDOU, S. 1990. Adaptive, minimal routing in hypercubes. In *Proceedings of the 6th MIT Conference on Advanced Research in VLSI*. (W. J. Dally, eds.), MIT Press, Cambridge, Mass., 139–453.

KONSTANTINIDOU, S., AND SNYDER, L. 1991. Chaos router: Architecture and performance. In *Proceedings of the 18th Annual International Symposium on Computer Architecture*, (Toronto, Ont., Canada, May 27–30), ACM, New York, p. 212–221.

LANDIN, A., HAGERSTEN, E., AND HARIDI, S. 1991. Race-free interconnection networks and multiprocessor consistency. In *Proceedings of the 18th Annual International Symposium on Computer Architecture* (Toronto, Ont., Canada, May 27–30). ACM, New York, pp. 106–115.

LINDER, D., AND HARDEN, J. 1991. An adaptive and fault tolerant wormhole routing strategy for k-ary n-cubes. *IEEE Trans. Comput., C-40*, 1 (Jan.) 2–12.

NGAI, J. 1989. A framework for adaptive routing in multicomputer networks. Ph. D. dissertation, California Institute of Technology. Caltech-CS-TR-89-09. Pasadena, Calif.

NGAI, J., AND SEITZ, C. 1989. A framework for adaptive routing in multicomputer networks. In *Proceedings of the 1989 ACM Symposium on Parallel Algorithms and Architectures* (Sante Fe, N. M., June 18–21). ACM, New York, pp. 2–10.

LILLEVIK, S. 1990. Intel Corporation. Touchstone program overview. In *Proceedings of the 5th Distributed Memory Computers Conference*. American Mathematical Society, Providence, RI.

SEITZ, C. L. 1985. The cosmic cube. *Commun. ACM 28*, 1 (Jan.), 22–33.

SEITZ, C. L., ATHAS, W. C., FLAIG, C. M., MARTIN, A. J., SEIZOVIC, J., STEELE, C. S., AND SU, W. 1988. The architecture and programming of the Ametek Series 2010 multicomputer. In *Proceedings of the 3rd Conference on Hypercube Computers* (Pasadena, Calif., Jan. 19–20). ACM, New York, pp. 33–36.

RECEIVED APRIL 1993; REVISED JANUARY 1994; ACCEPTED JANUARY 1994