Adaptive Deadlock-Free Routing in Multicomputers Using Only One Extra Virtual Channel

Chien-Chun Su and Kang G. Shin
Real-Time computing Laboratory
Department of Electrical Engineering and Computer Science
The University of Michigan
Ann Arbor, MI 48109–2122
{ccsu,kgshin}@eecs.umich.edu

Abstract: *We present three protocols defining the relationship between messages and the channel resources requested: request-then-hold, request-then-wait, and request-then-relinquish. Based on the three protocols, we develop an adaptive deadlock-free routing algorithm called the 3P routing. The 3P routing uses shortest paths and is fully-adaptive, so messages can be routed via any of the shortest paths from the source to the destination. Since it is a minimal or shortest routing, the 3P routing guarantees the freedom of livelocks.*

The 3P routing is not limited to a specific network topology. The main requirement for an applicable network topology is that there exists a deterministic, minimal, deadlock-free routing algorithm. Most existing network topologies are equipped with such an algorithm. In this paper, we present an adaptive deadlock-free routing agorithm for n-dimensional meshes by using the 3P routing. The hardware required by the 3P routing uses only one extra virtual channel as compared to the deterministic routing.

1 Introduction

Distributed-memory, MIMD (multiple-instruction-multiple-data) multicomputers usually consist of a large number of nodes, each with its own processor and local memory. These nodes use an interconnection network to exchange data and synchronize with one another. Thus, the performance of a multicomputer depends strongly on network latency and throughput.

There are two types of message routing: (1) *deterministic routing* that uses only a single path from source to destination, and (2) *adaptive routing* that allows more freedom in selecting message paths. Most commercial multicomputers use deterministic routing because of its deadlock freedom and ease of implementation. However, adaptive routing can reduce network latency, increase network throughput, and tolerate component failures. But the flexibility of adaptive routing may introduce deadlocks and/or livelocks. A deadlock occurs when a message waits for an event that will never happen. In contrast, a livelock keeps a message moving without progressing toward its destination.

A routing algorithm is said to be *minimal* if the number of hops of the routing path between two nodes is minimum and every hop brings the message closer to its destination. A minimal, fully-adaptive routing algorithm allows the message to be routed via any of the shortest paths between its source and destination. In this paper, we present a new fully-adaptive, deadlock-free, and minimal routing algorithm by using only one extra virtual channel as compared to the deterministic routing. This algorithm is based on the wormhole routing, which has higher transmission efficiency and requires less buffers than other switching methods. For the proposed routing algorithm, we assume that a deadlock-free, minimal routing function exists even if this extra virtual channel is not used. This assumption is reasonable because many popular topologies are equipped with such a routing function, e.g., e-cube routing for the hypercube, xy routing for the mesh, and the virtual-channel routing [2] for the torus.

Several adaptive, deadlock-free routing algorithms have been proposed in recent years. In [2], Dally and Seitz proposed the concept of virtual channel to develop deadlock-free routing algorithms. Virtual channels are logical abstractions that share the same physical link. They are time-multiplexed over a single physical link, so one separate queue must be maintained in a node for each virtual channel. Virtual channels are used to remove the cycles in a channel-dependency graph, thus providing deadlock freedom in message transmissions. However, the algorithms in [2] are deterministic.

The work described in this paper was supported in part by the National Science Foundation under Grant MIP–9203895. The opinions, findings, and recommendations expressed in this publication are those of the authors, and do not necessarily reflect the views of the NSF

In [6], Linder and Harden extended the concept of virtual channel to multiple, virtual interconnection networks that provide adaptability, deadlock-freedom and fault-tolerance. Each link is shared by many virtual channels, and the number of virtual channels used depends on how many virtual networks are needed. These virtual channels can be divided into several groups or virtual networks. Message passing inside a virtual network is deadlock-free and messages are constrained to travel through virtual networks in a defined order. When the message is blocked in a virtual network, it can keep going forward via another virtual network, thereby increasing routing adaptability. The chief disadvantage of this method is that many virtual channels (in general, an exponential function of the network dimension) are required, e.g., a k-ary n-cube needs $2^{n-1}(n+1)$ virtual channels.

In [4], Glass and Ni proposed partially-adaptive routing algorithms for 2D and 3D meshes without adding physical or virtual channels. They first investigated the possible deadlock cycles on 2D and 3D-meshes, then proposed some prohibited turns of these cycles to prevent deadlocks. However, if the minimal routing is required, then there exists only a single routing path for at least a half of source-destination pairs. Because this algorithm cannot route messages along any of shortest paths in the network, it is called partially-adaptive routing. Also, the higher the network dimension, the more source-destination pairs, each with only a single routing path, will result. The authors of [5] extended the same concept to n-dimensional meshes, k-ary n-cubes and hypercubes. In k-ary n-cubes, if $k > 4$, nonminimal routing should be used, thus taking more hops than needed and possibly introducing livelocks.

In [1], Chien and Kim proposed a planar-adaptive routing algorithm which limits the routing freedom to two dimensions at a time. The reduced freedom makes it possible to prevent deadlocks with only a fixed number of virtual channels, which is independent of network dimension. The hardware overhead is much less than that of the algorithm in [6].

In this paper, we propose an adaptive, deadlock-free, and livelock-free routing algorithm with less hardware overhead (except for partially-adaptive algorithms) and better adaptability than others. Section 2 presents three protocols that will be used to propose an adaptive deadlock-free routing algorithm. Section 3 describes the proposed routing algorithms for n-dimensional meshes. The proposed algorithm is compared with several other adaptive algorithms. The paper concludes with Section 4.

2 New Protocols for Adaptive, Deadlock-free Routing

In message-passing multicomputers, a message may be broken into one or more packets for transmission. According to the property of wormhole routing, a packet contains one or more flow-control digits (flits). The first flit of a packet (head flit) has to build the transmission path between the source and destination. Each flit of a packet following the head flit advances as soon as the preceding flit moves along (i.e., flits pipelining) and gets blocked when the required channel resources are unavailable.

A simple examination of the protocol for requesting channel resources in multicomputers leads to two cases:

1. Request-then-wait : While the requested channels are held by the other packets, the requesting packet will block and wait for the channels to be available.

2. Request-then-hold : While the requested channels are available, the packet will hold the channel and route the flits forward.

Now, we propose to add a third protocol:

3. Request-then-relinquish : Once a packet fails to get its requested channels, it terminates the request and does *not* wait for these channels either. In other words, no packet waits for a channel using this protocol.

Based on the above three protocols, we propose a new adaptive deadlock-free routing algorithm. First, we assume that:

1. The interconnection network can be divided into two virtual interconnection networks, VIN_1 and VIN_2, where VIN_1 supports deadlock-free minimal routing and VIN_2 uses one virtual channel (called *extra virtual channel*) to share the bandwidth of a physical link with the channels in VIN_1. The channels of VIN_2 are used to enhance routing adaptability.
2. Each virtual channel in VIN_1 is assigned a channel number. Also, packets requesting for the virtual channels in VIN_1 should obey strict increasing or decreasing order of channel numbers.
3. Only request-then-hold and request-then-relinquish protocols are used on the extra virtual channel (VIN_2), i.e., a requesting packet

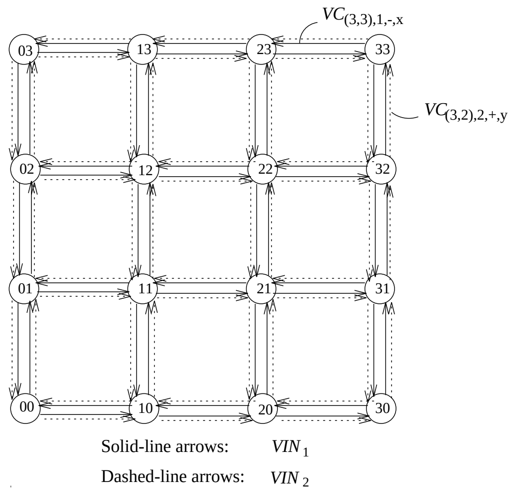

Figure 1: Two virtual interconnection networks in a 2-D mesh.

never waits for the extra virtual channel. Also, only request-then-wait and request-then-hold protocols are used on the virtual channels in VIN_1.

Fig. 1 shows VIN_1 (solid-line arrows) and VIN_2 (dashed-line arrows). In this case, each directed physical link is shared by two virtual channels. In case VIN_1 needs to use two virtual channels over each directed link (e.g., k-ary n-cubes), a total of three virtual channels are required on each physical link. The proposed adaptive routing algorithm, which can be applied to various topologies, is described as follows.

1. A requesting packet first checks if any extra virtual channel is available. Also, the minimum distance between the current location of the head flit and the packet's destination must be reduced after taking this hop (i.e., minimal routing). If these two conditions hold, then use the request-then-hold protocol else use the request-then-relinquish protocol, and search for the extra virtual channel of a different dimension. Repeat this search until the packet finds a channel of VIN_2 or no suitable channel can be found.
2. If no suitable channel of VIN_2 can be found in Step 1, then the packet waits for the virtual channel of VIN_1 according to an increasing or decreasing order of channel numbers.

For convenience, this proposed algorithm is called the $3P$ (three protocols) routing.

Theorem 1: The $3P$ routing is deadlock-free.

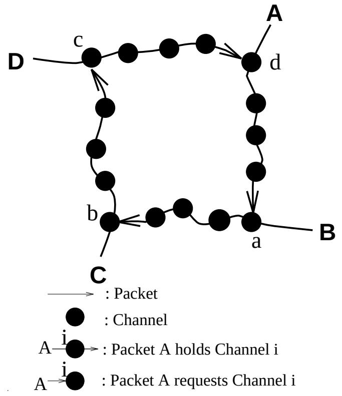

Figure 2: A deadlock of four packets.

Proof: We prove by contradiction. Assume there exists a deadlock for the proposed routing algorithm, then we can get a channel-dependency graph with a circular wait among the channel resources as shown in Fig. 2, where the arrow lines represent the packets and the black circles indicate the channels requested or held. This is a deadlock of four packets; one can extend this proof to those cases with more packets involved.

According to the proposed routing algorithm, the extra virtual channels never uses the request-then-wait protocol, so the channels a,b,c,and d in the graph are virtual channels of VIN_1. Since the deadlock-free routing in the virtual channels of VIN_1 should obey a strict order (say increasing) of requested channel numbers, the relationships among channel a, b, c, and d are summarized as:

$$\begin{array}{l} \text{a} < \text{b for packet B} \\ \text{b} < \text{c for packet C} \\ \text{c} < \text{d for packet D} \\ \text{d} < \text{a for packet A.} \end{array}$$

Thus, a<b<c<d<a, a contradiction. Therefore, no deadlock exists. □

In fact, the channels between a and b, b and c, c and d, d and a can be the channels of VIN_1 or VIN_2. If we remove or add a channel between two blocked channels, this change does not affect the deadlock freedom. In the next section, we apply this routing algorithm to n-dimensional meshes.

3 Application and Comparison

Before discussing the details of adaptive routing algorithms for different topologies, it is convenient to label each output virtual channel (VC) of a node with $VC_{node,VIN,direction,dimension}$, where *node* specifies which processing node to be used; $VIN = 1(2)$ indicates VIN_1 (VIN_2); $direction = +(-)$ represents positive (negative) direction; and *dimension* specifies which dimension to be used. For example, in Fig. 1, $VC_{(3,3),1,-,x}$ represents the output channel of node (3,3) of VIN_1 in the negative direction of dimension x.

Due to the limited space, we only present an adaptive, deadlock-free routing algorithm for n-dimensional meshes by using the $3P$ routing. For other topologies, such as k-ary n-cubes, C-wrapped hexagonal meshes [7] and so on, a similar method can be used, though the virtual channels required in VIN_1 may be different. An n-dimensional mesh consists of $k_0 \times k_1 \times \cdots \times k_{n-2} \times k_{n-1}$ nodes, where $k_i \geq 2$ is the number of nodes along dimension i. Each node X is represented by n coordinates, $(x_0, x_1, \ldots, x_{n-2}, x_{n-1})$, where $0 \leq x_i \leq k_i - 1$, $0 \leq i \leq n-1$. Two nodes X and Y are neighbers if and only if $x_i = y_i$ for all i, $0 \leq i \leq n-1$, except one, say j, such that $y_j = x_j \pm 1$. Thus, each node has from n to $2n$ neighbors, depending on its location in the mesh [4]. If X and Y are neighbors, then the channel of dimension i at node X is in positive direction when $x_i = y_i - 1$, or in negative direction when $x_i = y_i + 1$.

The xy routing [4] in meshes can be extended to n-dimensional meshes (called *extended xy* routing). That is, all packets should follow the dimension order, $0 \rightarrow 1 \rightarrow \ldots \rightarrow n-2 \rightarrow n-1$. The *extended xy* routing is deadlock-free, minimal and deterministic. Therefore, based on the *extended xy* routing, the proposed $3P$ routing can be applied to n-dimensional meshes. Based on the assumption of $3P$ routing, we need two virtual interconnection networks (VINs): VIN_1 supports *extended xy* routing and VIN_2 is used to enhance adaptability. Therefore, the bandwidth of each link needs to be shared by two virtual channels.

Adaptive routing algorithm for n-dimensional meshes:

Input: Source Node S = $(s_0, s_1, \ldots, s_{n-1})$;
Destination Node D = $(d_0, d_1, \ldots, d_{n-1})$;
Current Intermediate Node C
$= (c_0, c_1, \ldots, c_{n-1})$;
Initial: Routing Tag $R = (r_0, r_1, \ldots, r_{n-1})$
$= (d_0 - s_0, d_1 - s_1, \ldots, d_{n-1} - s_{n-1})$;

Step 1. Update R; /*$r_i := r_i + 1$ or $r_i := r_i - 1$*/
Step 2. If (R==0), flits arrive at destination;
Step 3. If (((any $r_i > 0$) && ($VC_{C,2,+,i}$ available)) || ((any $r_i < 0$) && ($VC_{C,2,-,i}$ available))),
if $r_i > 0$ then send packet via $VC_{C,2,+,i}$
else send packet via $VC_{C,2,-,i}$;
/*i can be chosen randomly or by any selection function */
Step 4. $i = min\{j : r_j \neq 0\}$;
/*Find the lowest dimension $i \exists r_i \neq 0$. */
Step 5. If $r_i > 0$ then request $VC_{C,1,+,i}$
else request $VC_{C,1,-,i}$;

In Step 1, the routing function used to update R is that (1) if the flit comes from $VC_{neighbor,*,+,i}$ then $r_i := r_i - 1$ and (2) if the flit comes from $VC_{neighbor,*,-,i}$, then $r_i := r_i + 1$, where * means "don't care." Since the routing tags can be used to specify the routing paths, it is not necessary to carry the source and destination addresses in the packets. In Step 3, request-then-hold and request-then-relinquish protocols are used to request the virtual channels of VIN_2. In Step 5, request-then-hold and request-then-wait protocols are used to request the channels of VIN_1.

Theorem 2: The $3P$ routing on n-dimensional meshes is minimal and deadlock-free. □

In n-dimensional meshes, the number of shortest paths for a source–destination pair is $(|r_0| + |r_1| + \cdots + |r_{n-1}|)!/|r_0|!|r_1|! \cdots |r_{n-1}|!$. Because the $3P$ routing on n-dimensional meshes can route packets via *any* available virtual channel of VIN_2 along shortest paths, it is fully-adaptive. The proposed algorithm only needs two virtual channels over each physical link. We can apply the methods proposed in [6] (abbreviated as the L-H algorithm in Fig. 3) to n-dimensional meshes, but it requires 2^{n-1} virtual channels. The planar-adaptive routing algorithm [1] (C-H algorithm in Fig. 3), which is not fully-adaptive, needs three virtual channels. The partially-adaptive routing algorithms in [4, 5] (G-N algorithm in Fig. 3) do not add any virtual channel, but they are not fully-adaptive.

Fig. 3 shows a table that compares the virtual-channel requirements of the $3P$ routing and other adaptive deadlock-free routing algorithms for several topologies. The proposed $3P$ routing requires only one additional virtual channel as compared to the deterministic routing. Furthermore, it is minimal and fully-adaptive. It can also be used to construct an adaptive deadlock-free routing algorithm for C-wrapped H-meshes [7]. The number of virtual channels required by the $3P$ routing on H-meshes is the same as that of k-ary n-cubes.

The $3P$ routing turns out to have a flavor somewhat similar to the one proposed in [3], although both have been developed independently. However, the method proposed in [3] dealt only with adaptive routing. The fault-tolerant $3P$ routing can be derived by changing the protocols of links around the faulty nodes or links. We will report the details of fault-tolerant routing in a forthcoming paper.

Topologies or properties \ Algorithms	Determ.	L-H	C-K	G-N	3P
2D mesh	1	2	2	1	2
Torus	2	6	4	#	3
n-D mesh	1	2^{n-1}	3	1	2
k-ary n-cube	1	$2^{n-1}(n+1)$	6	#	3
Hypercube	1	2^{n-1}	3	1	2
H-mesh	2	*	*	*	3
adaptability	no	fully	planar	partial	fully
minimal	yes	yes	yes	yes##	yes

*: Not mentioned in the published paper.
#: If k<5, then use 1 virtual channel, otherwise use nonminimal routing.
##: The class of k-ary n-cubes uses minimal routing only if k<5.

Figure 3: Comparison between virtual-channel requirements and routing properties for adaptive deadlock-free routing algorithms.

4 Conclusion

Communication efficiency is one of the most important factors to consider when designing a multicomputer system. The interprocessor communication speed strongly depends on the routing strategy, processor and data-link speed, and network topology. In this paper, we proposed three protocols of wormhole routing defining the relationship between messages/packets and channel resources: request-then-hold, request-then-wait, and request-then-relinquish. Based on these three protocols, an adaptive deadlock-free routing algorithm (the $3P$ routing) is proposed, which is then applied to various network topologies. Specifically, we presented adaptive routing algorithms by applying the $3P$ routing for n-dimensional meshes. We compare the $3P$ routing with the existing adaptive routing algorithms. For n-dimensional meshes and hypercubes, the $3P$ routing requires two virtual channels on each physical link. For k-ary n-cubes and C-wrapped hexagonal meshes, it requires three virtual channels. As compared to the deterministic routing, such as the xy routing, *extended xy* routing, e cube routing, and virtual channel routing [2], the $3P$ routing requires only one additional virtual channel regardless of network size/dimension. In addition to its deadlock freedom, the $3P$ routing is also minimal and fully-adaptive. The minimal routing ensures livelock-freedom. The fully-adaptiveness enables messages/packets to be transmitted via any of the shortest paths. Either the existing adaptive routing algorithms are just partially-adaptive, or require an excessive amount of hardware. By contrast, the proposed $3P$ routing uses little additional hardware despite its advantage of being fully-adaptive and minimal.

There are two directions of future work for the $3P$ routing. First, we need to implement the $3P$ routing with a compact hardware design. Second, it is necessary to evaluate the performance improvement by using the $3P$ routing.

References

[1] A. A. Chien and J. H. Kim, "Plannar-adaptive routing: Low-cost adaptive networks for multiprocessors," *in 19th Annual International Symposium on Computer Architecture*, pp. 268–277, 1992.

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

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

[4] C. J. Glass and L. M. Ni, "Adaptive routing in mesh-connected networks," *in Proceedings of the 1992 International Conference on Distributed Computing Systems*, pp. 12–19, 1992.

[5] C. J. Glass and L. M. Ni, "The turn model for adaptive routing," *in Proceedings of the 1992 International Symposium on Computer Architecture*, pp. 278–287, 1992.

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

[7] K. G. Shin, "HARTS: A distributed real–time architecture," *IEEE Computer*, vol. 24, no. 5, pp. 25–34, May 1991.