A Trail-Based Deadlock-Free Routing Scheme in Wormhole-Routed Networks

Jie Wu
Department of Computer Science and Engineering
Florida Atlantic University
Boca Raton, FL 33431

Abstract

This paper focuses on deadlock-free routing in wormhole-routed networks. A trail-based model is proposed to support deadlock-free routing in any networks with node degree no less than four. Basically, given a graph G representing a interconnection network, a walk in G is a finite sequence of edges. The trail-based model generalizes the trip-based model proposed earlier and it eliminates the need for virtual channels. Fundamentals of this model, including the necessary and sufficient condition of constructing a special trail for deadlock-free routing are investigated. The potential of this model is illustrated by applying it to meshes and hypercubes.

Key words: *deadlock, hypercubes, interconnection networks, meshes, wormhole routing*

1 Introduction

In a computer system, deadlocks arise when members of a group of processes which hold resources are blocked indefinitely from access to resources held by other processes within the group. Formally, a deadlock can arise if and only if the following four conditions hold simultaneously [4]:

- *Mutual exclusion*: no resource can be shared by more than one process at a time.
- *Hold and wait*: there must exist a process that is holding at least one resource and is waiting to acquire additional resources that are currently being held by other processes.
- *No preemption*: a resource cannot be preempted.
- *Circular wait*: there is a cycle in the wait-for graphs.

To avoid deadlock, it suffices to break one or more of the above four conditions. In this paper, we study the deadlock-free routing problem in wormhole-routed networks, where each source wants to send a message to some destination nodes. Routing represents the most important communication pattern. In a parallel/distributed system, the design and performance of a communication mechanism depends on several characteristics of the network architecture such as [3] *network topology, routing algorithm*, and *switching strategy*. The popular network topologies include 2-D mesh (Intel Paragon), 3-D mesh (MIT J-machine), 3-D torus (Cray T3D), and hypercube (Ncube Ncube-2). A system is called wormhole-routed if its switching technique used is *wormhole routing*, in which a message is divided into a number of *flits* that are pipelined through the network. One important feature of wormhole routing is the following: If there is no contention among messages for network resources the network latency is almost insensitive to the distance between the source and the destination. Because of the above feature, many nonminimal deadlock-free routing algorithms have been proposed for wormhole-routed systems.

Path-based routing [2] and [1] is an example of nonminimal routing algorithms. In this routing algorithm, the duplex-channel model is used, that is, each channel is bidirectional. This algorithm is based on finding a Hamiltonian path in the network. Then all the routings follow this selected Hamiltonian path (in both directions). Clearly, circular waiting is avoided, where each routing corresponds to a process and each edge corresponds to a resource, and there is no starvation; that is, each node is reachable from any other nodes. Recently, Tseng, Panda, and Lai [7] generalized the path-based solution and proposed a new *trip-based model*. This model can be applied to any network of arbitrary topology as long as it remains connected. The model is supported by adding a certain number of virtual channels to each physical channel and this number is derived depending on the trip.

In this paper, we propose a trail-based routing scheme which further generalizes the path-model without using virtual channels. The underlying channel model used is half-duplex, that is, messages can be flowed to either direction, but not both at the same time. We show that the trip-model can be applied to torus and hypercube networks. Fundamentals of this model, including the necessary and sufficient condition of constructing a special trail for deadlock-free routing are investigated.

2 Trail-based Deadlock-Free Routing

A parallel/distributed system is represented by an undirected system graph G [6]. More formally, G is defined to be a pair $(V(G), E(G))$, where $V(G)$ is a nonempty finite set of elements called vertices (also called nodes), and $E(G)$ is a finite family of unordered pairs of elements of $V(G)$ called edges (also called links). Normally, each vertex represents a processor and each edge is a communication link connecting two processors.

Definition 1: *Given any graph G, a walk in G is a finite sequence of edges of the form*

$$v_o \rightarrow v_1 \rightarrow v_2 \rightarrow \ldots \rightarrow v_{m-1} \rightarrow v_m.$$

It is clear that a walk has the property that any two consecutive edges are either adjacent or identical; however, an arbitrary sequence of edges of G which has this property is not necessarily a walk. The concept of a walk is too general for our purpose, so we shall impose some further restrictions.

Definition 2: *A walk in which all the edges are distinct is called a trail; if, in addition, the vertices $v_o, v_1, v_2, \ldots v_{m-1}, v_m$ are distinct (except, possibly, $v_0 = v_m$), then the trail is called a path. A path or trail is closed if $v_o = v_m$.*

In the path-based routing scheme, a Hamiltonian path which passes exactly once through each vertex of G is constructed. Because each channel is duplex, each ordered pair (v, u), where $v \neq u$ appears in the Hamiltonian path once and only once. A pair (v, u) is said to be in a path (or trail) if u is reachable from v using only edges in the path (or trail). Clearly, there is no circular wait. Therefore, path-based routing is deadlock free.

The condition for the existence of a Hamiltonian path is too strong. Moreover, if each channel is half-duplex, then the existence of a Hamiltonian path is not sufficient. Because, for any two vertices v and u, either (v, u) or (u, v) is not on the path.

In the proposed trail-based routing, a trail is constructed and a routing uses only those edges that appear in the trail. To ensure that each (ordered) pair of vertices (v, u) appears in the trail. We have the following necessary condition.

Theorem 1: *To ensure that each (ordered) pair of source and destination appears in a trail, except for one vertex which appears only once, all the other vertices should appear at least twice.*

Proof: If there are two vertices, say v and u, each of which appears only once in the trail. Then either (v, u) or (u, v) is not in the trail. □

By graph theory, any graph in which each node has even node degree (≥ 4) has a trail in which each node appears at least twice. Also, any graph where there are more than three nodes that have degree less than 4 does not have such a trail.

Corollary: *The trail-based routing cannot be applied to a 2-D mesh network where each channel is half-duplex.*

Proof: In a 2-D mesh, there are four nodes (in four corners) whose node degree is two. □

Note that two appearances of each node is just a necessary condition but not a sufficient condition. Consider a partial trail in the following, where number i in each superscript of a node represents the ith appearance of this node in the trail.

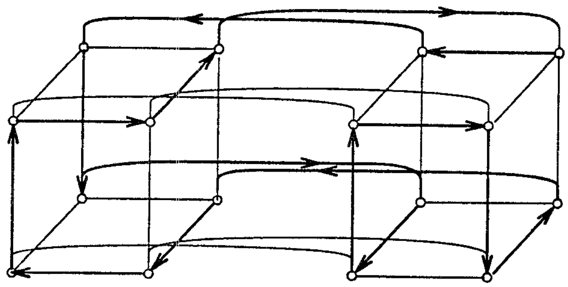
$$v_i^{(1)} \rightarrow v_i^{(2)} \rightarrow v_j^{(1)} \rightarrow v_j^{(2)}$$

Suppose both v_i and v_j appear only twice in the trail. Clearly, the pair (v_j, v_i) cannot be in this trail. Therefore, the necessary and sufficient condition for a given trail is the following:

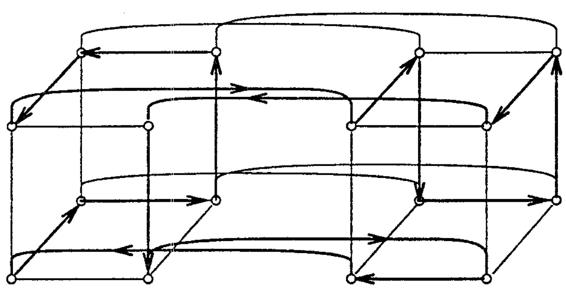
Theorem 2: *For any given node v_i, there is at least one appearance of any other node to the left of the right-most appearance of v_i and there is at least one appearance of any other node to the right of the left-most appearance of v_i.*

Actually, any two consecutive Hamiltonian paths meets the conditions in Theorem 2. In two consecutive Hamiltonian paths, each node appears exactly twice and all the nodes appear exactly once before the second appearance of any node. Note that when two Hamiltonian paths are merged into one, if the last node of one path is the same node of the first node of another path, then this node appears only once. Normally, two consecutive Hamiltonian paths requires a stronger condition than the one in Theorem 2. For example,

$$v_1^{(1)} \rightarrow v_4^{(1)} \rightarrow v_1^{(2)} \rightarrow v_3^{(1)} \rightarrow v_2^{(1)} \rightarrow$$



(a)



(b)

Figure 1: Two edge-disjoint Hamiltonian circuits in a 4-cube

$$\rightarrow v_1^{(3)} \rightarrow v_3^{(2)} \rightarrow v_4^{(2)} \rightarrow v_2^{(2)} \rightarrow v_3^{(3)}$$

meets the conditions in Theorem 2 but it fails the condition for two consecutive Hamiltonian paths. However, the following result show that if each node can appear no more than twice in a trail, then the condition for two consecutive Hamiltonian paths is equivalent to the one in Theorem 2.

Theorem 3: *If each node can appear no more than twice in a trail, then the condition for two consecutive Hamiltonian paths is a necessary and sufficient one.*

Proof: We first prove that all nodes appear exactly once in the trail before the second appearance of any node. By this we have the first Hamiltonian path. The fact is proved by contradiction. Suppose node v appear twice before the first appearance of node u. Because, each node can appear at most twice, there is no appearance of v after the first and the second appearances of u; therefore, pair (u, v) is not in the trail.

Now we prove that the rest of trail forms another Hamiltonian path. (The first element of the second Hamiltonian path can overlap with the last element of the first Hamiltonian path.) Again, we prove this fact by contradiction. Suppose there is no appearance of u in the rest of the trail, where u is not the last element in the first Hamiltonian path. Select any node v whose first appearance is after the first appearance of u. Clearly, (v, u) is not in the trail. □

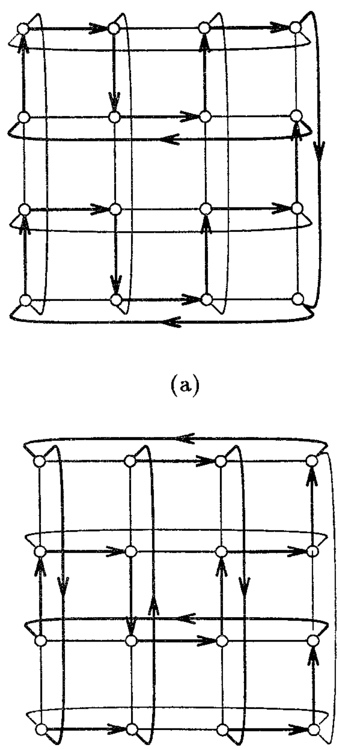
3 Examples

In this section, we apply the proposed trail-based routing to two popular network topologies: 2-D torus and n-dimensional hypercube. Two consecutive Hamiltonian paths are constructed by first identifying two *Hamiltonian circuits*. In a Hamiltonian path, if the first and last elements are the same then it is called a Hamiltonian circuits. Clearly, two consecutive Hamiltonian paths can be easily derived by appropriately removing two edges of two given Hamiltonian circuits, one from each Hamiltonian circuit to break the loop.

An n-dimensional hypercube (n-cube) [5] consists of $N = 2^n$ nodes and $n(N/2)$ links. $\{0, 1\}^n$ is the set of nodes where nodes u and v are connected if and only if u and v differ in exactly one bit. Figure 1 (a) and (b) show two edge-disjoint Hamiltonian circuits in a 4-cube.

The general way to construct edge-disjoint Hamiltonian circuits in an n-cube, $n \geq 4$, is as follows:

- Divide the given n-cube into two $(n-1)$-cubes along dimension n.
- Construct two Hamiltonian circuits, one from each of the $(n-1)$-cubes.
- Combine two edge-disjoint Hamiltonian circuits, one from each of two $(n-1)$-cubes, to form a Hamiltonian circuit in n-cube. This can be done by removing one edge to break the loop in each circuit and then add two edges along dimension n to connect two broken circuits.
- Combine the remaining two Hamiltonian circuits to form a Hamiltonian circuit in the n-cube.
- Once two edge-disjoint Hamiltonian circuits are constructed in the n-cube. Two consecutive Hamiltonian paths are derived by removing two edges from two edge-disjoint Hamiltonian circuits in the n-cube.



(a)

(b)

Figure 2: Two edge-disjoint Hamiltonian circuits in a 4×4 torus.

Figure 2 (a) and (b) show two edge-disjoint Hamiltonian circuits in a 4×4 torus. Two consecutive Hamiltonian paths can be derived from them straightforwardly.

4 Conclusions

In this paper, we have proposed a trail-based model to support deadlock-free routing. The objective is to find a trail that contain (v, u) and (u, v) for any v and u. We have proved that under the restriction of at most two appearances of each node the necessary and sufficient condition is to find a path that combines two consecutive Hamiltonian paths. This approach has been illustrated by its application to two popular network topologies: $2-D$ torus and n-dimensional hypercube. The problem of identification of a trail in a given graph that meet the necessary and sufficient condition without any constrain on the number of appearances of each node is still an open problem. Certain *Eulerian* graphs meet the condition. In a Eulerian graph G, there exists a closed trail which includes every edge of G; such a trail is then called an *Eulerian trail.* G is *semi-Eulerian* if we remove the restriction that the trail must be closed. The problem seems to be the one between the problem of identifying Eulerian graphs and the problem of identifying of Hamiltonian graphs.

References

[1] X. Lin, P. K. McKinley, and L. M. Ni. Performance evaluation of multicast wormhole routing in 2D-mesh multicomputers. *Proc. of Int'l Conf. on Parallel Processing.* I, 1991, 435-442.

[2] X. Lin and L. M. Ni. Deadlock-free multicast wormhole routing in multicomputer networks. *Proc. of Int'l Symp. on Computer Architecture.* 1991, 116-124.

[3] L. M. Ni and P. K. McKinley. A survey of routing techniques in wormhole networks. *Computer.* 26, (2), Feb. 1993, 62-76.

[4] J.L. Peterson and A. Silberschata. *Operating System Concepts.* Addison-Wesley Publishing Co., 1985.

[5] Y. Saad and M. H. Schultz. Topological properties of hypercubes. *IEEE Transactions on Computers.* 37, (7), July 1988, 867-872.

[6] R. J. Wilson. *Introduction to graph theory.* Longman Scientific and Technical, 1985.

[7] Y.C.Tseng, D. K. Panda, and T. H. Lai. A trip-based multicasting model in wormhole-routed networks with virtual channels. accepted to appear in *IEEE Transactions on Parallel and Distributed Systems.*