Multicasting on Wormhole-Routed Symmetric Networks with Hamiltonian Cycle Model

Nen-Chung Wang[1], Cheng-Pang Yen[2], and Chih-Ping Chu[2]
[1]*Department of Computer Science and Information Engineering*
National United University, Miao-Li 360, Taiwan, R.O.C.
Email: ncwang@nuu.edu.tw
[2]*Department of Computer Science and Information Engineering*
National Cheng Kung University, Tainan 701, Taiwan, R.O.C.
Email: chucp@csie.ncku.edu.tw

Abstract

In this paper, we first introduce a new hamiltonian cycle model for exploiting the features of symmetric networks. Based on this model, we propose two efficient multicast routing algorithms, uniform multicast routing algorithm and fixed multicast routing algorithm, in symmetric networks with wormhole routing. The proposed multicast routing algorithms utilizes channels uniformly to reduce the path length of message worms, making the multicasting more efficient in symmetric networks. We present two symmetric networks, the torus and star graph, to illustrate the superiority of the proposed schemes. Simulations are conducted to show that the proposed routing schemes outperform the previous scheme.

Keywords: Hamiltonian cycle, multicast, parallel computing, symmetric networks, wormhole routing.

1. Introduction

Multicomputer contains ensembles of computational nodes, with each node having its own processors, local memory, and other peripherals. Multicast is a collective communication operation for multicomputers. A single source sends the same message to numerous other destinations. Multicast communications can be denoted as a tuple (s, K), where s is the source node, $K = \{d_1, d_2, \ldots, d_n\}$ is the set of destination nodes, and n is the number of destination nodes.

Lin and Ni [6] derived a deadlock-free path-based multicast wormhole routing algorithm for direct networks. They proposed the new routing function based on the hamiltonian path embedded in the network to prevent cycle dependence. This path-based routing function is designed so that messages always acquire channels in the order that appears in the hamiltonian path to avoid deadlock. This hamiltonian-path-based multicast routing algorithm has trigger many other path-based multicast routing algorithms for diverse networks [2, 5]. The symmetric network is a special class of direct networks. It has the property that the network viewed from any vertex in the network looks the same [4]. With this property the congestion problem can be minimized because the load will be distributed uniformly through all of the vertices.

In this paper, we will study the deterministic path-based wormhole multicast routing problem on direct networks with symmetric network topologies. The focus is mainly on some popular networks, such as torus and star graph networks. The hamiltonian cycle model is proposed and two multicast routing algorithms are developed based on this model.

The rest of this paper is organized as follows. Preliminaries are presented in Section 2. In Section 3, we propose two multicast routing algorithms for bidirectional symmetric networks based on the hamiltonian cycle model. In Section 4, we apply the hamiltonian cycle model to two popular symmetric networks, the torus and star graph networks. Section 5 shows the simulation results. Finally, conclusions are given in Section 6.

2. Preliminaries

2.1. System Model

We first describe the system model required for this work. We assumed that the direct network topology is symmetrical with a hamiltonian cycle. We also assumed that the direct network adopts all-port wormhole switching and supports intermediate reception (IR) [3]. The wormhole switching mechanism is now widely used in interconnection networks, in both direct and indirect networks. We also assumed that the router is all-port. The all-port model specifies that the routers are able to relay multiple messages simultaneously provided that each incoming message has a unique outgoing channel leading to a neighboring node [7]. The IR allows a router to deliver an incoming message to the local host while simultaneously forwarding it to another router. In this manner, a single worm may be routed through several destinations, depositing a copy of the message at each destination. Such messages are called multidestination messages. All of the above mechanisms are commonly used in modern direct networks.

Multicomputers are usually interconnected via a direct network. The direct network topology is usually modeled as an undirected graph that defines how the nodes are interconnected by channels [7]. In the undirected graph $G = (V, E)$, the vertex set V represents the set of processors in the multicomputer and the edge set E defines the set of communication links in the direct network. Without loss of generality, G is assumed to be a connected graph. Each edge in E of the graph contains two opposite directed physical communication links. We used the notation $pch(u, v)$ to denote the physical channel going from u to v in the direct network. That is to say, each undirected edge $e(u, v) \in E$ in G, where $u, v \in V$, contains two opposite physical channels $pch(u, v)$ and $pch(v, u)$.

2.2. Hamiltonian Cycle Model

2.2.1. Labeling in the Hamiltonian Cycle

Hamiltonian cycle refers to a path existing in a graph that starts and ends at the same vertex and includes every vertex in the graph exactly once. In this paper, we focus on a direct network with a hamiltonian cycle. In the hamiltonian cycle model, the first step is to determine the hamiltonian cycle in the network graph. Although determining a hamiltonian cycle in any given graph is a NP-complete problem, there are many researches that presented strategies for embedding hamiltonian paths into popular network topologies, such as tori and star graphs. Now for a hamiltonian graph $G = (V, E)$ and $|V| = N$, we suppose that $\delta = (v_0, v_1, \dots, v_{N-1}, v_N, v_0)$ is a hamiltonian cycle in the graph G. According to the order of the vertex in the hamiltonian cycle δ, we can assign each vertex in the graph a label. The label of the vertex $v_i \in V$ is denoted as $\ell(v_i)$, where $\ell(v_i) = i$ is a natural number. That is, the hamiltonian cycle starts at the node labeled 0, and goes following the nodes with labels 1, 2, … , to the node with label $N-1$ consecutively, then returns the node labeled 0.

2.2.2. Boundary Links and Common Links

After assigning each node a label, we can divide every physical communication link in the network into two categories: a boundary link and a common link. The common link can be divided into two types: a general common link and a shortcut common link. It is easy to define the link with $|\ell(u)-\ell(v)|=1$ as a general common link. However, we do not want all of the links with $|\ell(u)-\ell(v)|>1$ to be boundary links. Some links designed as common links for shortcut transmissions. If the link with $|\ell(u)-\ell(v)|>\lceil N/2 \rceil$, we set the link as a boundary link, otherwise ($1<|\ell(u)-\ell(v)|\le\lceil N/2 \rceil$) we set the link as a shortcut common link.

2.2.3. Channel Networks

The network partitioning strategy is fundamental to our multicast routing algorithms. After assigning each node a label and defining the boundary link and common link in the network, we can divide the network into two subnetworks: a high-channel network and a low-channel network. The high-channel network contains all of the directional common channels with the nodes labeled from low to high numbers and the directional boundary channels with nodes labeled from high to low numbers. The low-channel network contains all of the directional common channels with the nodes labeled from high to low numbers and the directional boundary channels with nodes labeled from low to high numbers. After partitioning the network into two subnetworks, it is easy to see that every physical communication link lies in one and only one subnetwork, a high-channel network or low-channel network. Each of the two subnetworks has an independent set of physical links in the network.

2.2.4. Use of Virtual Channels

Two sets of virtual channels are used to prevent deadlock in the high-channel and low-channel subnetworks in our hamiltonian cycle model. Every common link in each subnetwork is multiplexed into p-channel and q-channel. The boundary link is not multiplexed into the two virtual channels. Only q-channel is required on the boundary link. We used the notation $c^{\alpha}_{u,v}$ to represent the virtual channel leaving node u to the node v, and belonging to type α of the virtual channel, where $\alpha \in \{ph, qh, pl, ql\}$. The ph, qh, pl, and ql represent the p-channel in the high-channel network, the q-channel in the high-channel network, the p-channel in the low-channel network, and the q-channel in the low-channel network, respectively.

The hold-and-wait property of wormhole routing is particularly susceptible to deadlock. Deadlock can be prevented using some specific routing algorithms. When network resources, such as nodes, are ordered and resources are accessed according to a strictly monotonic order, circular waiting for resources will not occur and deadlock can be avoided [5].

3. Multicast Routing Algorithms

3.1. Routing Function

The routing function R for the hamiltonian cycle model can be viewed as $R: N \times T \times N \rightarrow C$, which maps triple (current node, input channel type, destination node) into the next channel, where N is the set of the nodes in the network, T represents the set of types of the input virtual channels, and C is the set of the virtual channels. Let $n^h(u)$ denote the set of neighboring nodes of node u in the high-channel network and $n^l(u)$ represent the set of neighboring nodes of node u in the low-channel network. The routing function for the hamiltonian cycle model is described as follows.

$R(u, \alpha, v) = c^{\beta}_{u,w}$, where

$$\ell(w) = \begin{cases} \max\{\ell(x) \mid x \in n^h(u) \text{ and } \ell(x) \le \ell(v)\}, \text{if } \ell(u) < \ell(v) \text{ and } \alpha \in \{ph, qh\} \\ \max\{\ell(x) \mid x \in n^h(u) \text{ and } \ell(x) \le \ell(v)\}, \text{if } (\exists x \in n^h(u) \Rightarrow \ell(x) \le \ell(v)), \\ \qquad \ell(u) > \ell(v), \text{ and } \alpha \in \{ph, qh\} \\ \max\{\ell(x) \mid x \in n^h(u) \text{ and } \ell(x) > \ell(v)\}, \text{if } (\forall x \in n^h(u) \Rightarrow \ell(x) > \ell(v)), \\ \qquad \ell(u) > \ell(v), \text{ and } \alpha \in \{ph, qh\} \\ \min\{\ell(x) \mid x \in n^l(u) \text{ and } \ell(x) \ge \ell(v)\}, \text{if } \ell(u) > \ell(v) \text{ and } \alpha \in \{pl, ql\} \\ \min\{\ell(x) \mid x \in n^l(u) \text{ and } \ell(x) \ge \ell(v)\}, \text{if } (\exists x \in n^l(u) \Rightarrow \ell(x) \ge \ell(v)), \\ \qquad \ell(u) < \ell(v), \text{ and } \alpha \in \{pl, ql\} \\ \min\{\ell(x) \mid x \in n^l(u) \text{ and } \ell(x) < \ell(v)\}, \text{if } (\forall x \in n^l(u) \Rightarrow \ell(x) < \ell(v)), \\ \qquad \ell(u) < \ell(v), \text{ and } \alpha \in \{pl, ql\} \end{cases}$$

$$\beta = \begin{cases} ph, \text{if } \alpha = ph \text{ and } (u, w) \text{ is not a boundary link} \\ qh, \text{if } \alpha = qh \text{ or } (\alpha = ph \text{ and } (u, w) \text{ is a boundary link}) \\ pl, \text{if } \alpha = pl \text{ and } (u, w) \text{ is not a boundary link} \\ ql, \text{if } \alpha = ql \text{ or } (\alpha = pl \text{ and } (u, w) \text{ is a boundary link}). \end{cases}$$

A message is designed to route along only one subnetwork, the high-channel or low-channel network. Each subnetwork is connected. That is, there is at least one path between any two nodes in the subnetwork. Regardless if a message is routed along the high-channel or low-channel network, it is assigned a *p*-channel prior to crossing the boundary. It is assigned *q*-channel after crossing the boundary for it's remaining traverse.

3.2. Message Preparation Algorithms

In this subsection, we will propose two message preparation algorithms for the hamiltonian cycle model. Both of these message preparation algorithms are dual-path. That is, they will generate two outgoing messages to implement a multicast. These dual-path routing algorithms allow two packets outgoing from the source node simultaneously. The input for these message preparation algorithms is the set of destination nodes that the source node will communicate with. As we described before, the multicast routing algorithms for our hamiltonian cycle model have dual-paths. Therefore the output of both the message preparation algorithms are the two destination subsets, D^h and D^l. The two destination node subsets are partitions of the original destination set. D^h is the destination list in the header of the message M^h assigned to route along the high-channel network. Meanwhile, D^l is the destination list in the header of message M^l, which is assigned to route along the low-channel network. These message preparation algorithms are aimed to balance the routing path length and make multicasting more efficient.

3.2.1. Message Preparation Algorithm for Uniform Multicast Routing

The message preparation algorithm for uniform dual-path multicast routing easily balances the paths on which the two messages are routed. The destination nodes are partitioned into two subsets containing a near equal number of destination nodes according to their positions in the hamiltonian cycle. Let ϕ be the list contains the source node and the destination nodes. First, ϕ is sorted according to the label of nodes in ascending order. Next, ϕ is rotated so that the source node is the first element in the list. The source node is then discarded from ϕ. Subsequently, ϕ is partitioned into two subsets, D^h and D^l. D^h contains first $\lceil n/2 \rceil$ elements in ϕ and D^l contains the other elements in ϕ with reverse order. The multicast communication is denoted as (s, K). Suppose that $K = \{d_1, d_2, \dots, d_n\}$ is the set of destination nodes, where n is the number of destination nodes.

3.2.2. Message Preparation Algorithm for Fixed Multicast Routing

The message preparation algorithm concept for the fixed dual-path multicast routing involves restricting the maximal path length that two messages are routed. We give a pivot node to the source node. We assumed that the source node is s, the number of nodes is N, and the pivot node is p. If $\ell(s) < \lceil N/2 \rceil$, we set the pivot node p as the node with label $\ell(s) + \lceil N/2 \rceil$, otherwise we set the pivot node as the node with label $\ell(s) - \lceil N/2 \rceil$. Let ϕ be the list that contains the source node and destination nodes. First, ϕ is sorted according to the label for the nodes in ascending order. Next, ϕ is rotated so that the source node is the first element in the list. The source node is then discarded from ϕ. The message preparation process is completed using one of two cases: $\ell(s) < \lceil N/2 \rceil$ or $\ell(s) \ge \lceil N/2 \rceil$. Subsequently, ϕ is partitioned into two subsets, D^h and D^l. If $\ell(s) < \lceil N/2 \rceil$, D^h contains elements in ϕ with labels are greater than $\ell(s)$ and lower than $\ell(s) + \lceil N/2 \rceil$ and D^l contains the other elements in ϕ in reverse order. If $\ell(s) \ge \lceil N/2 \rceil$, D^l contains elements in ϕ with labels are greater than $\ell(s) - \lceil N/2 \rceil$ and lower than $\ell(s)$ in reverse order and D^h contains the other elements in ϕ. The multicast communication is denoted as (s, K). Suppose that $K = \{d_1, d_2, \dots, d_n\}$ is the set of destination nodes, where n is the number of destination nodes.

4. Hamiltonian Cycle Model in Torus and Star Graph Networks

Embedding one graph into another graph is useful as this permits using an algorithm designed for the former graph on the latter graph. In this section, we will show how the hamiltonian cycle model is applied to some popular symmetric networks, such as the torus and star graph networks. The steps include constructing a hamiltonian cycle in the network, partitioning the network, and arranging the virtual channels in the physical links. After embedding the hamiltonian cycle model into the two host graphs, those two multicast routing algorithms can be applied in these networks to exploit the symmetry property of these two graphs.

4.1. Mapping Hamiltonian Cycle Model to Torus Networks

In this section, we will first give a formal definition of the torus and then construct the hamiltonian cycle and embedded it into the torus network. According to the label for each node in the network, the network is partitioned into two subnetworks. After network partitioning, the communication links are multiplexed into the virtual channels to avoid deadlock. Based on the hamiltonian cycle model, our multicast routing algorithms can work. Finally we present an example to show how the messages are multicast in the tours network with a hamiltonian cycle model.

A formal description of the torus network is given below. An n-dimensional torus is denoted as $k_0 \times k_1 \times \ldots \times k_{n-1}$, where k_i refers to k_i nodes along each dimension i for $0 \le i \le n-1$. Each node u in the torus is identified using n coordinates, $\sigma_{n-1}(u)$, $\sigma_{n-2}(u)$, … , $\sigma_0(u)$, where $0 \le \sigma_i(u) \le k_i - 1$ for $0 \le i \le n-1$. Two nodes u and v in the bidirectional torus network are neighbors if and only if $\sigma_i(u) = \sigma_i(v)$ for all i , $0 \le i \le n-1$, but one, where $\sigma_i(u) \pm 1 = \sigma_i(v) \bmod k_i$. A torus network is regular if $k_i = k_j$ for all $0 \le i, j \le n-1$.

A 2D torus network contains many hamiltonian cycles. We use the node labeling function in [6] to construct a hamiltonian cycle in the 2D torus network. The node labeling function will assign each node a unique number. For a 2D $m \times n$ torus the hamiltonian cycle starts at the node numbered 0, following the nodes with labels 1, 2, … , to the node with label $mn-1$ consecutively, then returning the node numbered 0. The hamiltonian cycle starts and ends at the node labeled 0. In order to construct a hamiltonian cycle on the 2D $m \times n$ bidirectional torus network, we assumed that m is even.

In the following, we give the node labeling function $\ell(u)$ for a 2D $m \times n$ torus network. We suppose m is even and u is a node in the torus network, where the coordinates of u is (x, y), $0 \le x \le m-1$, and $0 \le y \le n-1$.

$$\ell(u) = \begin{cases} y \times n + x, & \text{if } y \text{ is even} \\ y \times n + n - x - 1, & \text{if } y \text{ is odd} \end{cases}$$

After assigning each node a label in the torus network, all of the physical communication links in the torus network can be divided into two categories: boundary links and common links. In this node labeling strategy, all wrap-around links in dimension 0 are boundary links, while the other physical links are common links. In the $2 \times k$ torus network, the boundary and common link definition in Section 2.2.2 may not fit. This is because the $2 \times k$ torus network is a multi-graph network. However, using the above node labeling function in the $2 \times k$ torus network, we can define the boundary links as the wrap-around links in dimension 0 and the other physical links as common links.
The network can then be divided into two subnetworks: a high-channel network and a low-channel network. To prevent deadlock in the high-channel and low-channel subnetworks in the torus network, two sets of virtual channels are used in each subnetwork. Every common link in each subnetwork is multiplexed into p-channel and q-channel. The boundary link is not multiplexed into two virtual channels. Only the q-channel is required in the boundary link.

Let us use an example to demonstrate the uniform and fixed multicast routing algorithms in the 2D 4×4 torus network. There is a multicast communication (s, K) in the 2D 4×4 torus network, where $s = (3, 2)^{11}$ and $K = \{(0, 0)^0, (1, 0)^1, (0, 2)^2, (1, 1)^6, (0, 2)^8, (2, 2)^{10}, (3, 3)^{12}, (2, 3)^{13}, (0, 3)^{15}\}$. Notice that the label for each node is shown as a superscript to the node representation. As shown in Figs. 1(a) and 1(b), a uniform multicast routing algorithm is used to implement this multicast. The source node $(3, 2)^{11}$ applies the uniform message preparation algorithm to generate two messages M^h and M^l, where the M^h header contains the destination nodes $(3, 3)^{12}$, $(2, 3)^{13}$, $(0, 3)^{15}$, $(0, 0)^0$, $(1, 0)^1$, and the M^l header contains the destination nodes $(2, 2)^{10}$, $(0, 2)^8$, $(1, 1)^6$, $(2, 0)^2$. M^h is routed along the high-channel network with a path length of 6. After M^h goes across the boundary, it will use the q-channel for routing. M^l is routed along the low-channel network with a path length of 7. Therefore, the maximum path length is max(6, 7) = 7.

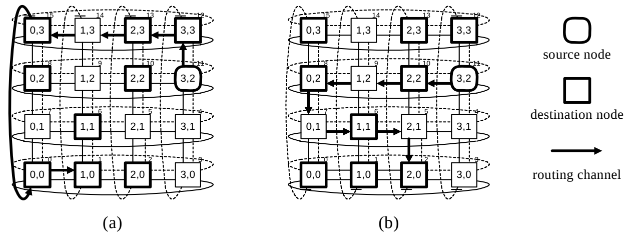

Fig. 1. A multicasting example in a 2D 4×4 torus network using uniform multicast routing: (a) high-channel network and (b) low-channel network.

As shown in Figs. 2(a) and 2(b), the same multicast example is used to demonstrate the fixed multicast routing algorithm. The source node $(3, 2)^{11}$ sends two messages M^h and M^l where the M^h header contains the destination nodes $(3, 3)^{12}$, $(2, 3)^{13}$, $(0, 3)^{15}$, $(0, 0)^0$, $(1, 0)^1$, $(0, 2)^2$, and the M^l header contains the destination nodes $(2, 2)^{10}$, $(0, 2)^8$, $(1, 1)^6$. M^h is routed along the high-channel network with a path length of 7. After M^h goes across the boundary, it will use the q-channel to continue its path. M^l is routed along the low-channel network with a path length of 5. Therefore, the maximum path length is max(7, 5) = 7.

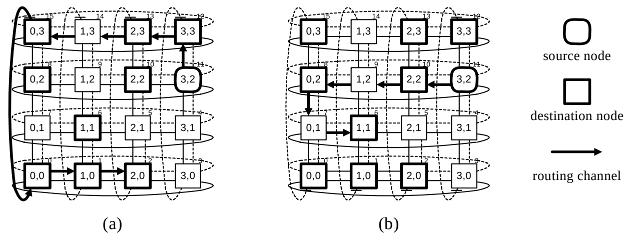

Fig. 2. A multicasting example in a 2D 4×4 torus network using fixed multicast routing: (a) high-channel network and (b) low-channel network.

4.2. Mapping Hamiltonian Cycle Model to Star Graph Networks

The star graph [1] is a special case of a Cayley graph. We introduce some preliminaries before giving a formal star graph definition. We first define the permutation. A permutation of n distinct symbols in the set $\{1, 2, \ldots, n\}$ is represented by $p = s_1s_2\ldots s_n$, where $s_i, s_j \in \{1, 2, \ldots, n\}$, $s_i \neq s_j$, for $i \neq j$, $0 \leq i, j \leq n$. A generator g_i is then defined. Given a permutation $p = s_1s_2\ldots s_n$, the generator g_i takes p as input and interchanges the symbol s_i with the symbol s_1 in p for $2 \leq i \leq n$. Thus, $g_i(p) = s_is_2\ldots s_{i-1}s_1\ldots s_n$. An undirected star graph with dimension n is denoted as $S_n = (V_n, E_n)$, where the set of vertices V_n is defined as $\{v \mid v = s_1s_2\ldots s_n, s_i, s_j \in \{1, 2, \ldots, n\}, s_i \neq s_j, \text{ for } i \neq j, 0 \leq i, j \leq n\}$ and the set of edges E_n is defined as $\{(v_1, v_2) \mid v_1, v_2 \in V_n, v_1 \neq v_2,$ such that $v_1 = g_i(v_2)$ for $2 \leq i \leq n\}$. In other words, the node in the star graph is represented by a permutation. Any two nodes v_1 and v_2 in the star graph are connected by undirected edges if and only if the permutation corresponding to the node v_2 can be obtained from that of v_1 by interchanging the symbol s_1 with s_i of v_1 for $2 \leq i \leq n$. A star graph has $n!$ nodes and node degree is equal to n - 1. We use the notation S_n or n-star to denote an n-dimensional star graph in this paper. The star graph superior features have been proven, such as low degree, small diameter, and symmetry.

Nigam et al. [8] have proposed an approach to embed the hamiltonian cycle in the star graph. In [2], Chen et al. derived a node labeling function for the star graph network based on the method proposed by Nigam et al. After assigning each node a label and defining the boundary and common links in the network, we can divide the network into two subnetworks: a high-channel network and a low-channel network. To prevent deadlock in the high-channel and low-channel subnetworks, two sets of virtual channels are used in each subnetwork. Every common link in each subnetwork is multiplexed into p-channel and q-channel. The boundary channel is not multiplexed into two virtual channels. Only the p-channel is required in the boundary channel.

Let us use an example to demonstrate the uniform and fixed multicast routing algorithms in the 4-star network. There is a multicast communication (s, K) in 4-star network, where $s = 1432^{17}$ and $K = \{2134^1, 3124^2, 2314^4, 1243^7, 4123^9, 3412^{12}, 3421^{19}, 2341^{21}, 3241^{22}\}$. Notice that the label for each node is shown as a superscript to the node representation. As shown in Figs. 3(a) and 3(b), a uniform multicast routing algorithm is used to implement this multicast. The source node 1432^{17} applies the uniform message preparation algorithm to generate two messages M^h and M^l, where the M^h header contains the destination nodes 3421^{19}, 2341^{21}, 3241^{22}, 2134^1, 3124^2, and the M^l header contains the destination nodes 3412^{12}, 4123^9, 1243^7, 2314^4. M^h is routed along the high-channel network with a path length of 9. After M^h goes across the boundary, it will use the q-channel to continue its path. M^l is routed along the low-channel network with a path length of 9. Therefore, the maximum path length is max(9,9) = 9.

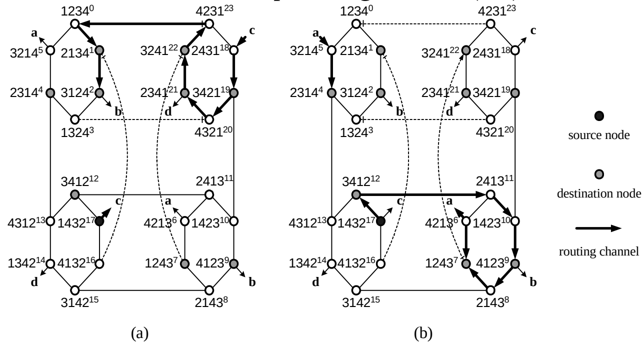

Fig. 3. A multicasting example in a 4-star network using uniform multicast routing: (a) high-channel network and (b) low-channel network.

As shown in Figs. 4(a) and 4(b), the fixed multicast routing algorithm is applied to the same multicast example. The source node 1432^{17} sends two messages M^h and M^l, where the M^h header contains the destination nodes 3421^{19}, 2341^{21}, 3241^{22}, 2134^1, 3124^2, 2314^4, and the M^l header contains the destination nodes 3412^{12}, 4123^9, 1243^7. M^h is routed along the high-channel network with a path length of 11. After M^h goes across the boundary, it will use the q-channel for routing. M^l is routed along the low-channel network with a path length of 6. Therefore, the maximum path length is max(11, 6) = 11.

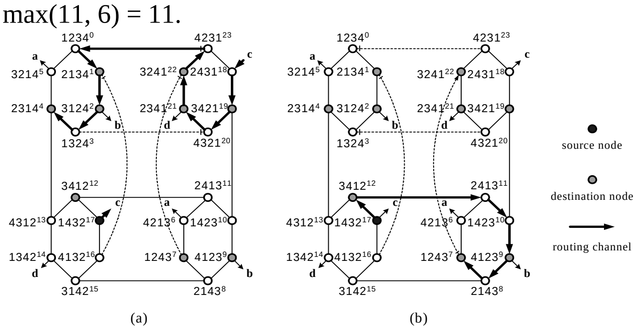

Fig. 4. A multicasting example in a 4-star network using fixed multicast routing: (a) high-channel network and (b) low-channel network.

5. Performance Study

We conducted some simulations for torus and star graph networks. For all simulation experiments, the following system parameters are used: the startup latency is 1 microsecond. The network latency is 25 nanoseconds. All of these parameters reflect the current trend in technology. The source and destination nodes were selected randomly in each experiment. Each experiment was repeated 1024 times with the average latency determined.

First, the system model for the simulation is in 2D 64×64 bidirectional torus networks with various message sizes. The multicast size is the number of destination nodes. The message length is the number of flits in a message. Fig. 5 shows the average communication latency for message length of 120 flits in the 2D 64×64 torus network. From Fig. 5, it is easy to see that the performance of the dual-path routing algorithm is worse than that of the uniform and fixed routing algorithms. The dual-path routing algorithm takes more time to implement the multicast than the uniform and fixed routing algorithms do. The two hamiltonian-cycle-based routing algorithms can take advantage of the wrap-around channels to balance two message worms and to reduce the path length of each message worm. The performance of the fixed and uniform multicast routing algorithms is almost the same in the 2D torus network.

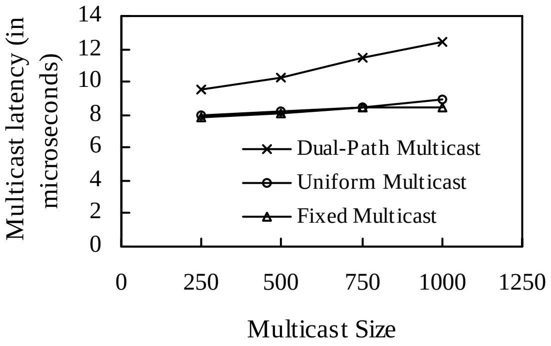

Fig. 5. Simulation in 2D torus network with message lengths of 120 flits.

Next, the proposed routing algorithms are applied to the star graph network. We compare these routing algorithms with the dual-path routing algorithm in star graph networks proposed by Chen et al. [2]. The system model for the simulation is in the 6-star network with various message sizes. The multicast size is the number of destination nodes. The message length is the number of flits in a message. Fig. 6 shows the average communication latency for the message length of 120 flits in the 6-star network. As shown in Fig. 6, the performance of the uniform and the fixed routing algorithms are better than that of the dual-path routing algorithm. This difference shows the different strategies to improve the performance of the multicast communication. This implies that the message preparation is the critical part of the multicast routing algorithm. Improvement of the message preparation is more effective to the performance of the multicast communication. The performance of fixed and uniform multicast routing algorithms is almost the same in the 6-star network.

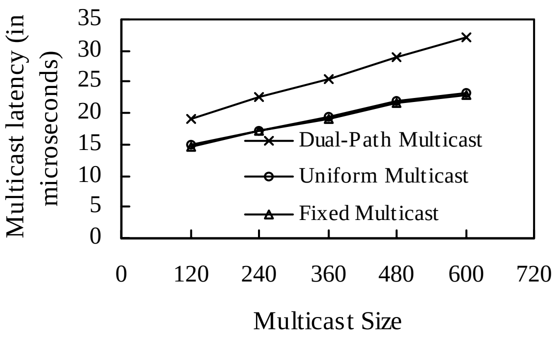

Fig. 6. Simulation in 6-star network with message length of 120 flits.

6. Conclusions

In this paper, a hamiltonian cycle model was proposed to provide an efficient multicast communication in symmetric networks. Two deadlock-free multicast routing algorithms were presented based on the hamiltonian cycle model. Both of these multicast routing algorithms have the advantage of reducing the number of traversed links for the messages to improve multicast performance. Simulation results showed that the performance of the hamiltonian-cycle-based multicast routing algorithms was better than the hamiltonian-path-based multicast routing algorithm in the corresponding torus and star graph networks.

Acknowledgments

This work was supported by the National Science Council of Republic of China under grants NSC-93-2218-E-324-001 and NSC-94-2213-E-324-025.

References

[1] S. B. Akers, D. Harel and B. Krishnamurthy , "The Star Graph: An Attractive Alternative to The n-cube," *Proceedings of the 1987 International Conference on Parallel Processing,* pp. 393-400, August 1987.

[2] T.-S. Chen, N.-C. Wang, and C.-P. Chu, "Multicast Communication in Wormhole-Routed Star Graph Interconnection Networks," *Parallel Computing,* Vol. 26, No. 11, pp. 1459-1490, October 2000.

[3] E. Fleury and P. Fraigniaud, "Strategies for Path-Based Multicasting in Wormhole-Routed Meshes," *Journal of Parallel and Distributed Computing,* Vol. 53, pp. 26-62, August 1998.

[4] M. J. Flynn, "Some Computer Organizations and Their Effectiveness," *IEEE Transactions on Computers,* C-21, pp. 948-960, September 1972.

[5] X. Lin, P. K. McKinley, and L. M. Ni, "Deadlock-Free Multicast Wormhole Routing in 2D Mesh Multicomputers," *IEEE Transactions on Parallel and Distributed Systems,* Vol. 5, No. 8, pp. 793-804, October 1994.

[6] X. Lin and L. M. Ni, "Deadlock-Free Multicast Wormhole Routing in Multicomputer Networks," *Proceedings of the 18th Annual International Symposium on Computer Architecture,* pp. 116-125, 1991.

[7] L. M. Ni and P. K. McKinley, "A Survey of Wormhole Routing Techniques in Direct Networks," *IEEE Computer,* Vol. 26, No. 2, pp. 62-76, February 1993.

[8] M. Nigam, S. Sahni, and B. Kirshnamurthy, "Embedding Hamiltonians and Hypercubes in Star Interconnection Graphs," *Proceedings of the 1990 International Conference on Parallel Processing,* Vol. 3, pp. 340-343, August 1990.