Multicast Communication on Wormhole-Routed Star Graph Interconnection Networks with Tree-Based Approach

Nen-Chung Wang
Dept. of Information and Communication Engineering
Chaoyang University of Technology
Taichung, Taiwan, R.O.C.
E-mail: ncwang@mail.cyut.edu.tw

Chih-Ping Chu
Dept. of Computer Science and Information Engineering
National Cheng Kung University
Tainan, Taiwan, R.O.C.
E-mail: chucp@csie.ncku.edu.tw

Abstract

Multicast is an important collective communication operation on multicomputer systems, in which the same message is delivered from a source node to an arbitrary number of destination nodes. The star graph interconnection network has been recognized as an attractive alternative to the popular hypercube network. In this paper, we address an efficient and deadlock-free tree-based multicast routing scheme for wormhole-routed star graph networks with hamiltonian path. In our proposed routing scheme, the router is with the input-buffer-based asynchronous replication mechanism that requires extra hardware cost. Meanwhile, the router simultaneously sends incoming flits on more than one outgoing channel. We perform simulation experiments with the network latency and the network traffic. Experimental results demonstrate that our proposed routing scheme outperform the previous approaches.

Keywords: Multicast, parallel computing, star graphs, tree-based routing, wormhole routing.

1 Introduction

Multicast is an important collective communication operation on multicomputer systems, in which the same message is delivered from a source node to an arbitrary number of destination nodes.

Strategies for multicasting can be classified as three approaches: unicast-based, path-based, and tree-based. In unicast-based multicast algorithms, a source node sends messages to its set of destinations by sending a sequence of separate unicast messages to each destination [8]. The unicast-based algorithms use one-to-one communication to achieve multicast, which requires startup latency in each intermediate node. The disadvantage of this approach lies in that significant transmission latency is resulted from the required number of communication startups for multicast. The path-based multicast algorithms allow a worm to contain multiple destination (*multidestination*) address in its header flits. They use a simple hardware mechanism to allow routers to absorb flits on internal channel (to the local processor) while simultaneously forwarding copies of the flits on output channels enroute to the remaining destinations [5, 7, 11, 12]. In this way, a message can be delivered to several destinations by a single worm that only need a single startup. The path-based multicast algorithms are highly inefficient because the network has to be traversed multiple times, and flits of the message have to be copied and forwarded by the network interface associated with the nodes [14]. For example, consider the hamiltonian path-based algorithms, the dual-path multicast routing requires only two startups to send a message to any set of destinations in a mesh, while the multipath multicast routing requires four startups but frequently uses shorter paths to all destinations [7].

Intuitively, a tree-based multicasting scheme requires shorter paths to reach the destinations and is thus more efficient than a path-based scheme. A potential problem with tree-based multicasts on wormhole-routed networks is that they can easily cause deadlocks, due to the interdependency between different tree branches [4]. Recently, as shown in the literature [6, 13, 15], many researches have been focused on the multicasting algorithms with tree-based routing. Tree-based algorithms attempt to deliver the message to all destinations in a single multi-head worm that splits at some routers and replicates the data on multiple output ports [6]. Data replication can be implemented with two approaches: synchronous replication [9] and asynchronous replication [9]. Synchronous replication requires that flits of a multidestination worm proceed in lock-step [6]. Thus, any branch of the multidestination worm that is blocked can block all other branches. Asynchronous replication allows that flits of different multidestination worm can progress independently through the network and bubble flits are inserted where necessary [6]. Since synchronous replication scheme requires complex signaling hardware at the routers, simple asynchronous replication scheme is preferred for a practical implementation. More recently, a deadlock-free input-buffer-based asynchronous replication mechanism [14] was proposed for implementing routers. This technique was shown to be effective in breaking the interdependency between tree branches. But, the mechanism requires extra hardware cost (the extra MUXs and the additional control logic).

The star graph [1, 2] interconnection network has been recognized as an attractive alternative to the popular hypercube network. Some reasones are its symmetric and hierarchical, and lower degree and smaller diameter as opposed to the hypercube. In this paper, we address an efficient and deadlock-free tree-based multicast routing scheme for wormhole-routed star graph networks with hamiltonian path. In tree-based routing, the destination set is divided into two destination subsets and then the multicasting is proceeded by two independent paths (one for high-channel routing and the other for low-channel routing) based on two disjoint subnetworks for concurrent transmission. For each independent path, the message is delivered to the destination subset with a single multidestination worm that splits at some routers and replicates the data on more than one output port. In our proposed scheme, for deadlock-free routing, the router also applies the input-buffer-based asynchronous replication mechanism [14]. We will show that our proposed tree-based scheme is superior to the previous approaches.

The rest of this paper is organized as follows. Preliminaries are presented in Section 2. In Section 3, we propose tree-based multicast routing scheme. Simulation results of these algorithms are presented in Section 4. Finally, concluding remarks are drawn in Section 5.

2 Preliminaries

2.1 System Model

In the following, we first introduce some definitions and notations related to the star graphs. A permutation of n distinct symbols from the set $\{1, 2, \cdots, n\}$ is represented as $p = s_1 s_2 \cdots s_n$, where $s_i, s_j \in \{1, 2, \cdots, n\}, s_i \neq s_j$ for $i \neq j, 1 \leq i, j \leq n$. Given a permutation $p = s_1 s_2 \cdots s_n$, let the *generator* g_i be the function of p that interchanges the symbol s_i with the symbol s_1 in p for $2 \leq i \leq n$. Thus, $g_i(p) = s_i s_2 \cdots s_{i-1} s_1 s_{i+1} \cdots s_n$. 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 $\{v | v = s_1 s_2 \cdots s_n, s_i, s_j \in \{1, 2, \cdots, n\}, s_i \neq s_j$ for $i \neq j, 1 \leq i, j \leq n\}$ and the set of edges E_n is defined as $\{(v_p, v_q) | v_p, v_q \in V_n, v_p \neq v_q$, such that $v_q = g_i(v_p)$ for $2 \leq i \leq n\}$.

In other words, any two nodes v_p and v_q are connected by an undirected edge if and only if the corresponding permutation to the node v_q can be obtained from that of v_p by interchanging the symbol s_i of v_p with the symbol s_1 of v_p for $2 \leq i \leq n$. We also use the notation S_n to represent an n-dimensional star graph, called n-star, in this paper. Notice that star graphs are edge and vertex symmetric. Moreover, S_n is a regular graph with degree $n-1$, $n!$ vertices, and $\frac{(n-1)n!}{2}$ edges. A 3-star and a 4-star are shown in Figure 1.

The interconnection network system is composed of nodes, each node is a computer with its own processor, local memory, and communication links; each link connects two neighboring nodes through network [7]. A common component of nodes in a new-generation multicomputer is

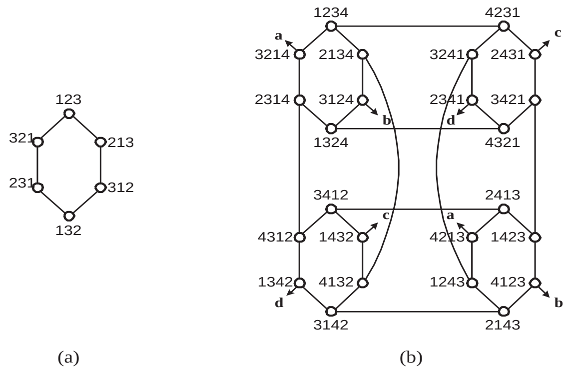

Figure 1: The topology of star graphs: (a) 3-star; (b) 4-star.

a *router*. It can handle the entering, leaving, and passing through the node of message. Figure 2 shows the architecture of a generic node. A router is usually connected to the local processor/memory by one or more pairs of internal channels. One channel of each pair is for input, the other for output. Several pairs of external channels connect the router to neighboring routers. The interconnection of external channels among routers defines the network topology.

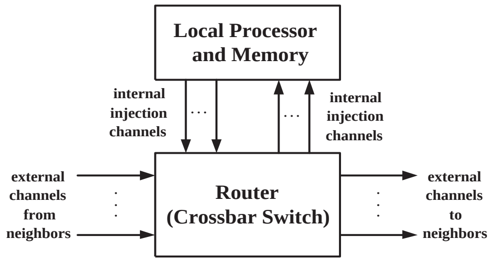

Figure 2: A generic node architecture.

We assume that a message coming into a router can be duplicated and delivered simultaneously to multiple output channels. We also assume that whenever a message header enters an input buffer, all the remaining flits are guaranteed to enter that buffer. The routers are implemented with the input-buffer-based asynchronous replication mechanism [14].

2.2 Path-Based Multicast Routing Model

In our previous work on wormhole star graph networks routing [3], we addressed a path-based routing model, derived a node labeling formula based on a single *hamiltonian path* (HP), and proposed four efficient deadlock-free multicast routing schemes: dual-path, shortcut-node-based dual-path, multipath, and proximity grouping. Generally, the dual-path scheme is simple and efficient. The multicasting in the dual-path routing includes two independent paths (toward high label nodes and low label nodes, respectively) and the next traversed node is the neighboring node with the label nearest to that of the next unvisited target node. The concept of the path-based routing model is described below.

2.2.1 Hamiltonian Paths and Channel Networks

The path-based routing method for meshes developed by Lin et al. [7] is based on a HP. In [3], we used the strategy in [10] to define a HP on the star graph. Because a star graph is embedded with more than one HP, the routing methods proposed in [3] is simply on basis of a specific HP of all possible HPs.

In an n-star, the number of nodes is $N = n!$ and each node s is with a label $\ell(s)$, where $0 \le \ell(s) \le N-1$ and $\ell()$ is the node labeling function [3]. The labeling of a 4-star based on a HP is shown in Figure 3. For example, in a 4-star, $\ell(1234) = 0$, $\ell(4213) = 6$, $\ell(4312) = 13$, $\ell(4231) = 23$, and so forth.

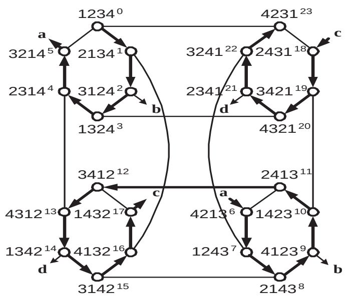

Figure 3: The labeling of a 4-star based on a hamiltonian path.

According to the node labels, we can construct a specific HP, i.e., from the node with label 0, following the nodes with labels $1, 2, \cdots$, to the node with label $N-1$. When node labeling is completed, we can divide the network into two subnetworks, *high-channel network* and *low-channel network*. The *high-channel network* contains all directional channels with nodes labeled from the lower to the higher, and the *low-channel network* contains all directional channels with nodes labeled from the higher to the lower. Then, a message routing can be performed along two legal paths, one along *high-channel network* and the other along *low-channel network*. An example showing the channel subnetworks of a 4-star is given in Figure 4(a) and Figure 4(b), respectively.

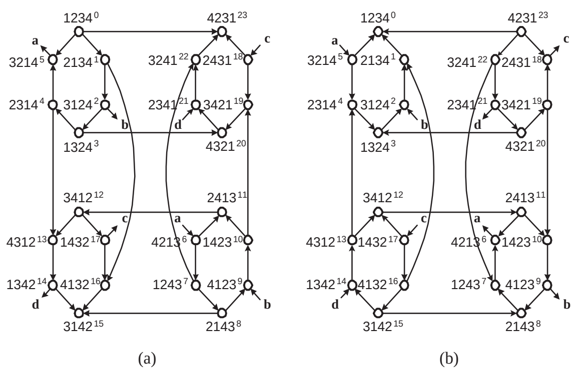

Figure 4: The channel networks of a 4-star: (a) high-channel network; (b) low-channel network.

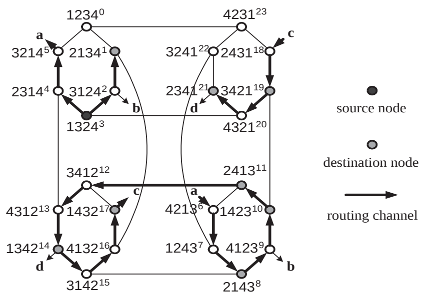

Figure 5: A sample multicast example using hamiltonian-path and dual-path routing.

2.2.2 Hamiltonian-Path and Dual-Path Multicast Routing

The unicast-based, the hamiltonian-path, and the dual-path routing strategies can be adopted in a lot of wormhole-routed interconnection networks. The unicast-based routing scheme uses one-to-one communication to achieve multicast, which requires startup latency in each intermediate node [8]. The disadvantage of this approach lies in that significant transmission latency is resulted from the required number of communication startup steps for multicast. In the hamiltonian-path routing, the source node sends the message to all destination nodes based on the constructed hamiltonian path. In this scheme, the multicast is divided into two submulticasts and that can be proceeded in parallel by two independent routing paths (one for high-channel routing and the other for low-channel routing). The disadvantage of this approach is that it always traverses nodes following the fixed path (hamiltonian-path) that requires more traverse links for multicast [7]. In the dual-path routing, the multicasting is similar to the hamiltomian-path routing except each router tries to find a shortcut node (the node with label closest to that of the next *unvisited target* node) for routing to reduce the average length of multicast paths [7].

A sample multicast using hamiltonian-path and dual-path routing is shown in Figure 5. The sample multicast is denoted as the multicasting set $R = \{\underline{1324}^3, 2134^1, 2143^8, 1423^{10}, 2413^{11}, 1342^{14}, 1432^{17}, 3421^{19}, 2341^{21}\}$, where the first element of R is the source node and the others are the destination nodes in arbitrary order. Notice that the source node is underlined, the label $\ell(u)$ of each node u in R is shown as a superscript to the node representation. In hamiltonian-path and dual-path routing, the multicasting set R can be completed by two submulticasting sets, R^h for high-channel routing and R^l for low-channel routing, i.e., $R^h = \{\underline{1324}^3, 2143^8, 1423^{10}, 2413^{11}, 1342^{14}, 1432^{17}, 3421^{19}, 2341^{21}\}$ and $R^l = \{\underline{1324}^3, 2134^1\}$. In R^h and R^l, the first elements are source nodes and the others are destination nodes with label values higher and lower than source nodes and in ascending and descending orders, respectively. In hamiltonian-path and dual-path routing as shown in Figure 5, the total number of channels traversed is 18+2=20, and the maximum routing distance is max(18,2)=18.

The hold-and-wait property of wormhole routing is particularly susceptible to deadlock, and thus most wormhole-routed systems avoid messages routing to reach cycles of channel dependency. Deadlock can be prevented by the routing algorithm. By ordering network resources, such as nodes, and accessing resources according to a strictly monotonic order circular wait for resources will not occur and deadlock can be avoided [7].

3 Tree-Based Multicast Routing

For the dual-path routing, the performance is unstable specially for large multicast sizes. That is, if the traversed node number of high-channel routing is nearly identical to that of low-channel routing, then dual-path routing performs very well; otherwise the performance of dual-path routing depends on longer traversed node number of either high-channel routing or low-channel routing. Therefore, we propose the tree-based routing with replication mechanism to promote the performance of the multicasting. Before we introduce the proposed routing algorithm, let us first define a routing functions RF.

Definition 1 (*The routing functions* RF). Let V, p, q, and $\ell()$ be the node set, the source node, the destination node of a star graph, and the node labeling function [3], respectively. The routing function RF, is defined to be $RF : V \times V \to V$ and $RF(p,q) = x$, and if $\ell(p) < \ell(q)$, then $\ell(x)$ = max$\{\ell(u)|\ell(p) < \ell(u) \leq \ell(q)$, and u is adjacent to $p\}$; if $\ell(p) > \ell(q)$, then $\ell(x)$ = min$\{\ell(u)|\ell(p) > \ell(u) \geq \ell(q)$, and u is adjacent to $p\}$.

The tree-based routing scheme includes four steps. First, the destination node set D is divided into two subsets, D^h and D^l, where every node in D^h has a higher label than that of the source node s, and every node in D^l has a lower label than that of the source node s according to the node labeling function. Then, the destination nodes in D^h are sorted according to the $\ell()$ values in ascending order and the destination nodes in D^l are sorted according to the $\ell()$ values in descending order, respectively. Third, we construct two messages, M^h and M^l, where M^h contains D^h as part of the header and M^l contains D^l as part of the header. Finally, the multicast is proceeded by the following two submulticasts in parallel: (1) the message M^h is sent to the nodes in D^h using tree-based high-channel routing based on subnetwork N^h, and (2) the message M^l is sent to the nodes in D^l using tree-based low-channel routing based on subnetwork N^l.

The message transmission in tree-based routing is proceeded according to the following rules.

Rule 1: For message M^h routing in high-channel subnetwork, each sending node finds the high neighboring node set P^h that contains the neighboring nodes with higher $\ell()$ values. Then, let w be the neighboring node in P^h with maximum $\ell()$ value. If w exists and is a destination, the router replicates the message and sends it by two disjoint paths. Otherwise, the router sends the message to the neighboring node which has $\ell()$ value that is the greatest but less than that of the first destination node.

Rule 2: For message M^l routing in low-channel subnetwork, each sending node finds the low neighboring node set P^l that contains the neighboring nodes with lower $\ell()$ values. Then, let w be the neighboring node in P^l with minimum $\ell()$ value. If w exists and is a destination, the router replicates the message and sends it by two disjoint paths. Otherwise, the router sends the message to the neighboring node which has $\ell()$ value that is the least but larger than that of the first destination node.

Rule 3: While the message visits a node, the router determines whether it is the first destination node. If so, it is removed from the destination nodes. Then, at this node, if the destination sets are not empty, the algorithm continues according to the Rule 1 or Rule 2.

The tree-based routing algorithm is shown in Algorithm 1, whereas the tree-based high-channel routing and tree-based low-channel routing algorithms are shown as Procedure 1 and Procedure 2, respectively. In the following, we use the same multicast example as the one used for the hamiltonian-path and dual-path routing to demonstrate the better multicast performance of the tree-based routing when compared with the hamiltonian-path and dual-path routing. In the sample multicast, the multicasting set R can be completed by two submulticasting sets, R^h and R^l, where $R^h = \{\underline{1324}^3, 2143^8, 1423^{10}, 2413^{11}, 1342^{14}, 1432^{17}, 3421^{19}, 2341^{21}\}$ and $R^l = \{\underline{1324}^3, 2134^1\}$. In R^h and R^l, the first elements are source nodes and the others are destination nodes with higher and lower label values than source nodes in ascending $\ell()$ and descending $\ell()$ value orders, respectively. R^h routes the message using high-channel routing based on subnetwork N^h. R^l routes the message using low-channel routing based on subnetwork N^l.

Algorithm 1: *The tree-based routing algorithm*
Input: Source node s, destination node set D, and node labeling function $\ell()$.
Step 1: // Destination-nodes partition
Divided D into two subsets, D^h and D^l, D^h contains all destination nodes with higher $\ell()$ values than source node s and D^l contains all destination nodes with lower $\ell()$ values than source node s.
Step 2: // Destination-nodes sorting
Sort the destination nodes in D^h according to the $\ell()$ values in ascending order. Sort the destination nodes in D^l according to the $\ell()$ values in descending order.
Step 3: // Message preparation
Construct two messages M^h and M^l, where M^h contains D^h as part of the header and M^l contains D^l as part of the header.
Step 4: // Routing in parallel
// The message M^h is sent to the nodes in D^h using tree-based
// high-channel routing based on subnetwork N^h.
// The message M^l is sent to the nodes in D^l using tree-based
// low-channel routing based on subnetwork N^l.
$Tree\_Based\_High\_Channel\_Routing(s, M^h)$
$Tree\_Based\_Low\_Channel\_Routing(s, M^l)$

Figure 6 shows the sample multicast example using tree-based routing. The detailed routing process of the sample multicast is shown in the Appendix. From Figure 6, the total number of channels traversed is 3+2+5+2+4+2 = 18, and the maximum routing distance from the source to a destination is max(3+max(2+max(5,2),4),2)=10. So, the total number of channels traversed and the maximum routing distance of tree-based routing are smaller than that of dual-path routing.

To verify the correctness of the tree-based routing algorithm, we derive the following lemmas and theorems.

Procedure 1:
$Tree\_Based\_High\_Channel\_Routing(s, M^h)$
// tree-based high-channel routing proceeds on subnetwork N^h
begin
For message M^h which contains D^h do
$c := s$
while $(D^h \neq \emptyset)$
// for every current node c, and next traversed destination node d
// each sending node finds neighboring nodes with higher $\ell()$ values
$P^h := \{u|\ell(u) > \ell(c)$, and u is adjacent to $c\}$
// find greatest $\ell()$ value in P^h
$\ell(w) := \max\{\ell(u)|u \in P^h\}$
// check whether the message needs to replicate or not
if (w exists) and ($w \in D^h$)
// the router replicates the message and sends it in parallel
$D^{h1} := \{u|u \in D^h$ and $\ell(u) \geq \ell(w)\}$
$D^{h2} := D^h - D^{h1}$
Let message M^{h1} contains D^{h1} as part of the header and message M^{h2} contains D^{h2} as part of the header
// The first path
traverse node w
$D^{h1} := D^{h1} - \{w\}$
$Tree\_Based\_High\_Channel\_Routing(w, M^{h1})$
// The second path
$Tree\_Based\_High\_Channel\_Routing(c, M^{h2})$
else // find next node x to traverse
// M^h routing along higher $\ell()$ value
// $x = RF(c, d)$, where x is the next traversed node
// and RF is the routing function
get the destination node d with least $\ell()$ value from D^h
$\ell(x) := \max\{\ell(u)|\ell(c) < \ell(u) \leq \ell(d)$, and u is adjacent to $c\}$
if (x = d)
// traverse node d and then remove it from D^h and message M^h
traverse node d
$D^h := D^h - \{d\}$
Remove d from message M^h
endif
$c := x$
endif
endwhile
end

Procedure 2:
$Tree\_Based\_Low\_Channel\_Routing(s, M^l)$
// tree-based low-channel routing proceeds on subnetwork N^l
begin
For message M^l which contains D^l do
$c := s$
while $(D^l \neq \emptyset)$
// for every current node c, and next traversed destination node d
// each sending node finds neighboring nodes with higher $\ell()$ values
$P^l := \{u|\ell(u) < \ell(c)$, and u is adjacent to $c\}$
// find least $\ell()$ value in P^l
$\ell(w) := \min\{\ell(u)|u \in P^l\}$
// check whether the message needs to replicate or not
if (w exists) and ($w \in D^l$)
// the router replicates the message and sends it in parallel
$D^{l1} := \{u|u \in D^l$ and $\ell(u) \leq \ell(w)\}$
$D^{l2} := D^l - D^{l1}$
Let message M^{l1} contains D^{l1} as part of the header and message M^{l2} contains D^{l2} as part of the header
// The first path
traverse node w
$D^{l1} := D^{l1} - \{w\}$
$Tree\_Based\_Low\_Channel\_Routing(w, M^{l1})$
// The second path
$Tree\_Based\_Low\_Channel\_Routing(c, M^{l2})$
else
// M^l routing along lower $\ell()$ value
// $x = RF(c, d)$, where x is the next traversed node
// and RF is the routing function
get the destination node d with greatest $\ell()$ value from D^l
$\ell(x) := \min\{\ell(u)|\ell(c) > \ell(u) \geq \ell(d)$, and u is adjacent to $c\}$
if (x = d)
// traverse node d and then remove it from D^l and message M^l
traverse node d
$D^l := D^l - \{d\}$
Remove d from message M^l
endif
$c := x$
endif
endwhile
end

Lemma 1. *For two arbitrary distinct nodes p and q in a star graph with a HP, the path from p to q selected according to the routing function RF always exists.*
Proof. Suppose p and q are two arbitrary nodes in a star graph, without loss of generality, it can be assumed that $\ell(p) < \ell(q)$. Let the node c represent the source node or the intermediate node located in between source node p and destination node q on HP. Assume the next traversed node is x, according to the routing function RF, $x = RF(c, q)$, where $\ell(x) = \max\{\ell(u)|\ell(c) < \ell(u) \leq \ell(q)$, and u is adjacent to $c\}$. So, x is on HP going from c to q (including q) and adjacent (connected) to c. Then, the path from p to q selected according to the routing function RF is $(y_0, y_1, \cdots, y_k)$, where $y_0 = p$, $y_j = RF(y_{j-1}, q)$ for $0 < j \leq k$, and $y_k = q$. Since all the nodes of the path are located on HP and in an order, the path from p to q selected according to the routing function RF always exists. ☐

Lemma 2. *The tree-based high-channel message routing, based on subnetwork N^h, in a star graph with a HP can always be completed.*
Proof. Based on Lemma 1, it is obvious. ☐

Lemma 3. *The tree-based low-channel message routing, based on subnetwork N^l, in a star graph with a HP can always be completed.*
Proof. Based on Lemma 1, it is obvious. ☐

Theorem 1. *The message routing using tree-based routing algorithm in a star graph with a HP can always be completed.*
Proof. The message routing using tree-based routing algorithm is proceeded by two submulticasts simultaneously. For one submulticast, the tree-based high-channel routing can be completed via high-channel subnetwork N^h. For the other submulticast, the tree-based low-channel routing can be completed via low-channel subnetwork N^l. According to Lemma 2 and Lemma 3, either tree-based high-channel or tree-based low-channel message routing can be completed. So, the message routing using tree-based routing algorithm can always be completed. ☐

Theorem 2. *The tree-based multicast routing is deadlock-free.*
Proof. At the source node, the tree-based algorithm divides the networks into two disjoint subnetworks N^h and N^l. Because $N^h \cap N^l = \emptyset$, the tree-based multicast routing is deadlock-free at each of the two subnetworks. Then, let us prove that messages delivered in subnetwork N^h are deadlock-free. Messages delivered in N^h can only take high-channels in N^h. At an intermediate node in

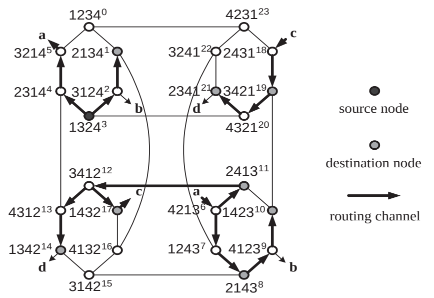

Figure 6: The sample multicast using tree-based routing.

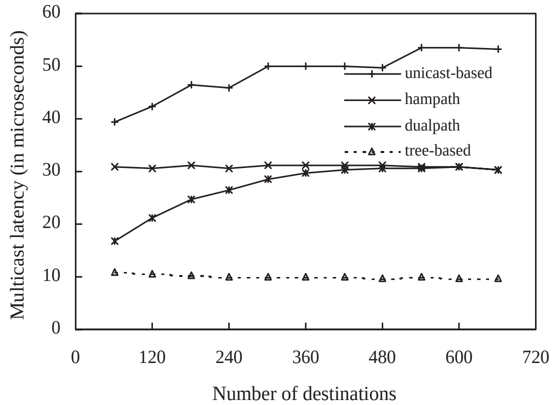

Figure 7: Multicast latency in a 6-star network with with 120 flits message length.

N^h, tree-based algorithm divides the destination set D^h into two disjoint destination subsets D^{h1} and D^{h2}, where $D^{h1} \cap D^{h2} = \emptyset$, so no cyclic dependency can be created among the channels in N^h. Furthermore, since the input-buffer-based asynchronous replication mechanism [14] is able to break the interdependency of different tree-branches at branch node, multiple messages delivered in N^h will not cause deadlocks. Similar proof can be applied to the sub-network N^l. This thus proves the theorem. □

In our proposed tree-based multicast routing algorithm, we use the channel subnetworks that have been described in previous section. Because the subnetworks are disjoint and acyclic, no cyclic resource dependency can occur [7]. Thus, the proposed routing algorithm developed based on those two subnetworks are deadlock-free.

4 Simulation Results

In this section, we shall present the performance of our proposed multicasting strategies by some simulation experiments. we first give some assumptions to the parameters of system architecture and our algorithm in our simulation. All simulation were performed for a 720-node (6-dimension) star graph network. The source node and destination nodes for each multicasting were randomly generated. The message startup latency t_s is 1.0 microsecond (550 nanoseconds for message sending latency, 450 nanoseconds for message receiving latency). The link propagation latency t_l is 5.0 nanoseconds. The router latency for handling multidestination messages t_r is 40.0 nanoseconds; however, it is set to 20.0 nanoseconds in unicast-based routing. For all of the multicasting on our simulation, the message length is assumed to be 120 flits.

Figure 7 present the performance of the various multicast schemes on a 6-star network. It is observed that, the hamiltonian-path, the dual-path, and the tree-based algorithms outperform the unicast-based algorithm. This is because the unicast-based algorithm is a multiple-phase multicasting that needs more startup latency for processing. In Figure 7, the performance of our proposed tree-based algorithm is superior to that of the unicast-based, the hamiltonian-path, and the dual-path algorithms. This is because the tree-based algorithm uses asynchronous replication mechanism for simultaneous transmission that efficiently reduces the communication latency.

In our simulations, multicast latencies were measured with a varying number of destinations. In general, for unicast-based and path-based routing, while the number of destinations increases, the communication latency always increases. However, for tree-based multicasting, simulation results in [6] demonstrated that message latency is independent of the number of destinations. Similarly, as shown in Figure 7, the communication latencies for our proposed tree-based algorithm are also irrelevant with the number of destinations. That is, the communication latencies are stable across different multicast sizes. This is because that the tree-based scheme has the advantage of simultaneous transmission in the split router. The advantage makes the communication latencies stable across different number of destinations.

5 Conclusions

In this paper, we propose an efficient tree-based multicast routing scheme for wormhole-routed star graph networks with hamiltonian path and present the performance of the proposed scheme in contrast with our previous work. In tree-based routing, the destination set is divided into two destination subsets and then the multicasting is proceeded by two independent paths (one for high-channel routing and the other for low-channel routing) based on two disjoint subnetworks for concurrent transmission. For each independent path, the message is delivered to the destination subset with a single multidestination worm that splits at some routers and replicates the data on more than one output port. In our routing scheme, the router is with the input-buffer-based asynchronous replication mechanism that requires extra hardware cost. The proposed routing scheme is proved to be deadlock-free. By the experimental results, our proposed tree-based scheme is superior to the previous approaches.

Acknowledgements

This work was supported by the National Science Council of Republic of China under Grant #NSC-91-2218-E-324-002.

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] S.B. Akers and B. Krishnamurthy, "A Group-Theoretic Model for Symmetric Interconnection Networks," *IEEE Trans. on Computers,* Vol. 38, No. 4, pp. 555-565, April 1989.

[3] 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.

[4] J. Duato, S. Yalamanchili, and L.M. Ni, *Interconnection Networks: An Engineering Approach,* Computer Society Press, 1997.

[5] K.-P. Fan and C.-T. King, "Turn Grouping for Multicast in Wormhole-Roued Mesh Networks Supporting the Turn Model," *The Journal of Supercomputing,* Vol. 16, No. 3, pp. 237-260, July 2000.

[6] R. Libeskind-Hadas, D. Mazzoni, and R. Rajagopalan, "Tree-Based Multicasting in Wormhole-Routed Irregular Topologies," *Proceedings of the Merged Twelfth International Parallel Processing Symposium and the Ninth Symposium on Parallel and Distributed Processing,* April 1998.

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

[8] P.K. McKinley, H. Xu, A.H. Esfahanianm, and L.M. Ni, "Unicast-Based Multicast Communication in Wormhole-Routed Networks," *IEEE Trans. on Parellel and Distributed Systems,* Vol. 5, No. 12, pp. 1252-1265, December 1994.

[9] L. Ni, "Should Scable Parallel Computers Support Efficient Hardware Multicast?," *Proceedings of the 1995 ICPP Workshop on Challenges for Parallel Processing,* pp. 2-5, August 1995.

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

[11] D.K. Panda, S. Singal, and R. Kesavan, "Multidestination Message Passing in Wormhole k-ary n-cube Networks with Base Routing Conformed Path," *IEEE Trans. on Parellel and Distributed Systems,* Vol. 10, No. 1, pp. 76-96, January 1999.

[12] D.F. Robinson, P.K. McKinley, and B.H.C. Cheng, "Path-Based Multicast Communication in Wormhole-Routed Unidirectional Torus Networks," *Journal of Parallel and Distributed Computing,* Vol. 45, No. 2, pp. 104-121, September 1997.

[13] R. Sivaram, D.K. Panda, and C.B. Stunkel, "Multicasting in Irregular Networks with Cut-Through Switches using Tree-Based Multidestination Worms," *Proceedings of the 2nd Parallel Computing, Routing, and Communication Workshop (PCRCW'97),* pp. 35-48, Atlanta, Georgia, June 1997.

[14] R. Sivaram, C.B. Stunkel, and D.K. Panda, "Implementing Multidestination Worms in Switch-Based Parallel Systems: Architectural Alternatives and their Impact," *IEEE Trans. on Parallel and Distributed Systems,* Vol. 11, No. 8, pp. 794-812, August 2000.

[15] Z. Zhou, W. Shi, and Z. Tang, "Efficient Multidestination Multicast on Regular Router-Based Networks," *Proceedings of the Fourth International Conference/Exhibition on High Performance Computing in Asia-Pacific Region (HPC Asia'2000),* Vol. 1, pp. 82-87, May 2000.