

Broadcasting Algorithms Based upon the Virtual Channel in the Star Graph Interconnection Network *

Tain-Shiang Harng and Chang-Biau Yang

Department of Applied Mathematics

National Sun Yat-sen University

Kaohsiung, Taiwan 80424

cbyang@math.nsysu.edu.tw

Abstract

The deadlock problem of a broadcasting algorithm in an interconnection network can be overcome by using the concept of virtual channels. In this paper, we first apply the virtual channel concept to propose four broadcasting algorithms on the star graph and the incomplete star graph. Then we consider about message redundancy in broadcasting, and two broadcasting algorithms without message redundancy on the star graph and the incomplete star graph respectively are proposed. They all require $\lfloor \frac{n+1}{2} \rfloor$ virtual channels for each physical link in the n -dimensional star graph. In the previous results, the best one is that a routing algorithm requires $\lfloor \frac{n+1}{2} \rfloor$ virtual channels for each physical link. And it is clear that the number of virtual channels required in a routing algorithm is no more than that in a broadcasting algorithm. Thus, our broadcasting algorithms obtain a good result in the number of virtual channels.

Key words: star graph, broadcasting algorithm, virtual channel, deadlock-free

1 Introduction

In an interconnection network, if a routing algorithm is adaptive, it can provide some flexible paths when a faulty link or a faulty node exists. On the other side, the defect is that deadlock might be caused more easily when more than one source node attempts to transmit message to other nodes. The deadlock problem in routing or broadcasting can be overcomed by using the concept of *virtual channels* [4, 5]. Virtual channels are logical channels which share a phys-

*This research work was partially supported by the National Science Council of the Republic of China under contract NSC 87-2213-E-110-004.

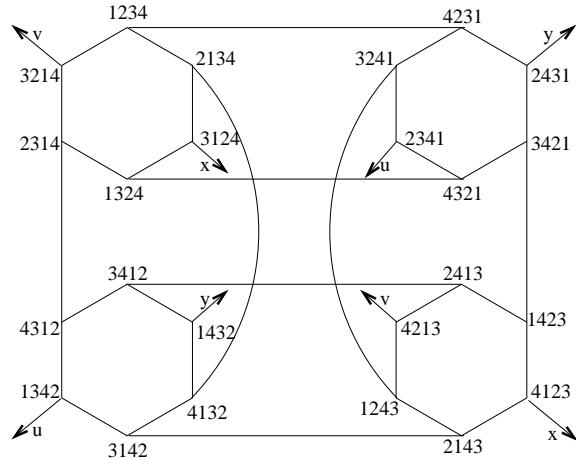


Figure 1: A 4-dimensional star graph, S_4 .

ical channel by using the time-division multiplexing method. Each virtual channel of one physical link to one node has its own queue. However, the more virtual channels are used, the larger buffers and bandwidth associated with each physical channel are needed, and then the cost for communication is increased. Thus, the goal of using this concept is to decrease the number of virtual channels.

An n -star graph [1], denoted as S_n , consists of $n!$ nodes labeled with $n!$ different permutations of n distinct symbols $(1, 2, \dots, n)$. g_i is defined as $g_i(p_1 p_2 \cdots p_n) = (p_i p_2 p_3 \cdots p_{i-1} p_1 p_{i+1} \cdots p_n)$, where $p_1 p_2 \cdots p_n$ is a permutation of $123 \cdots n$ and $2 \leq i \leq n$. Two nodes u and v are connected in S_n if and only if there exists a generator g_i such that $g_i(u) = v$. Note that $g_i(v) = u$ if $g_i(u) = v$. For example, Figure 1 shows a 4-star. In S_4 , $g_2(1234) = 2134$, $g_3(1234) = 3214$, and $g_4(1234) = 4231$.

In S_n , each node has degree $n - 1$. S_n has a low diameter $\lfloor \frac{3(n-1)}{2} \rfloor$ [1]. The star graph is both node

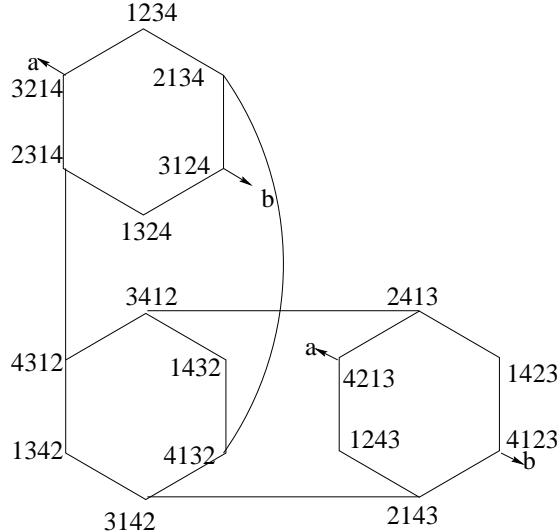


Figure 2: The $C_3(3)$ graph.

and edge symmetric [1, 2, 6, 11], and it is optimality faulty tolerant [1, 3]. It has some advantages over the hypercube: a lower degree, a small diameter, and a smaller average diameter, and has a simple fault tolerant routing algorithm [11], and a fault tolerant routing algorithm [7].

The C_{n-1} graph is a special kind of incomplete star graphs proposed by Latifi [8]. A $C_{n-1}(k)$ graph is a collection of k $(n-1)$ -stars, in which the nodes are labeled as $\{xxx\dots x(n)\}, \{xxx\dots x(n-1)\} \dots, \{xxx\dots x(n-k+1)\}$, where x is used to denote the variant digit of the node label in the C_{n-1} graph. A node in the C_{n-1} graph connects to its neighbors as the star graph except the one belonging to the nonexistent substar. For example, Figure 2 shows the $C_3(3)$ graph, consisting of 3 3-stars which have 18 nodes. In the graph, $k = 3$, the nodes are labeled as $xxx4$, $xxx3$, and $xxx2$.

Yang and Liu [12] proposed two wormhole routing algorithms on the star graph, S_n . Virtual channel is switched to another one if the polarity changes from negative to positive. The first one is minimal, fully adaptive, and deadlock-free and it requires $\lfloor \frac{3n+1}{4} \rfloor$ virtual channels per physical link. In the second algorithm, some restrictions are made on the routing path selection. Thus, it reduces the number of virtual channels to $\lfloor \frac{n+1}{2} \rfloor$. It is still minimal and deadlock-free. However, it becomes partially adaptive.

In this paper, we apply the virtual channel concept to propose four deadlock-free broadcasting algorithms on the star graph and the incomplete star graph. The first and the second are on the star graph

and the incomplete star graph respectively. Then we consider about message redundancy in broadcasting, and two broadcasting algorithms without message redundancy on the star graph and the incomplete star graph respectively are proposed. They all require $\lfloor \frac{n+1}{2} \rfloor$ virtual channels for each physical link in the n -dimensional star graph. In the previous results, the best one is that a routing algorithm [12] requires $\lfloor \frac{n+1}{2} \rfloor$ virtual channels for each physical link. And it is clear that the number of virtual channels required in a routing algorithm is no more than that in a broadcasting algorithm. Thus, our broadcasting algorithms obtain a good result in the number of virtual channels.

The rest of this paper is organized as follows. In Section 2, we shall define some notations used in this paper. From Section 3 through Section 6, we shall propose four broadcasting algorithms. And finally, a conclusion will be given in Section 7.

2 Definitions and Notations

In this section, we shall define some notations used in this paper. We use $A[j]$ to denote the j th symbol of a node A , where the left most symbol is counted as 1, i.e. $A[1]$ denotes the leftmost symbol. The *polarity* of on the link from node A to node B is defined to be positive if $A[1] < B[1]$; otherwise, it is negative. Note that the link polarity from A to B and the link polarity from B to A are complement to each other.

Function f_{p_i} is defined as $f_{p_i}(p_1 p_2 \dots p_n) = (p_i p_2 p_3 \dots p_{i-1} p_1 p_{i+1} \dots p_n)$, where $p_1 p_2 \dots p_n$ is a permutation of $123 \dots n$ and $2 \leq i \leq n$. In other words, if f_{p_i} is performed, symbol p_i is moved to the first position.

If node A is connected to node B with link g_i (i.e., $g_i(A) = B$), we can express it by function f as $f_{A[i]}(A) = B$.

3 Broadcasting in the Star Graph

Broadcasting is to transmit message from a source node to all other nodes. Mendiola et al. [9] proposed an optimal broadcasting algorithm on S_n . In this section, we shall propose our virtual channel approach based on Mendiola et al.'s optimal broadcasting. Our method makes some restrictions on the broadcasting sequence such that the broadcasting method is still optimal and only $\lfloor \frac{n+1}{2} \rfloor$ virtual channels is required for each physical link.

We shall briefly introduce Mendiola et al.'s optimal broadcasting approach. Since the star graph is a re-

cursive hierarchical structure, an S_n can be partitioned into $n S_{n-1}$'s by fixing the symbol on the rightmost position, denoted as $S_{n-1}^1, S_{n-1}^2, \dots, S_{n-1}^n$, where in S_{n-1}^i , the rightmost symbol is i . In each S_{n-1} , one node, called the *leader node*, is chosen to broadcast message to other nodes in S_{n-1} . Since the star graph is node symmetric, without loss of generality, the source node can be any node in S_n .

The broadcasting can be done by recursively applying the following 2-phase algorithm.

Phase 1: By the *recursive doubling scheme*, a *relay tree* is generated such that each symbol in $\{A[1], A[2], \dots, A[n-1]\}$ appears in the first symbols of the tree nodes exactly once. These tree nodes are called *relay nodes*.

Phase 2: By applying g_n (i.e. $f_{A[n]}$), the $n-1$ relay nodes transmit the message to the leader nodes of $S_{n-1}^{A[1]}, S_{n-1}^{A[2]}, \dots, S_{n-1}^{A[n-1]}$, and the original source node also plays the role of the leader node of $S_{n-1}^{A[n]}$.

Figure 3 shows an example for broadcasting in one iteration by the two-phase algorithm. After each leader node receives the original message, it plays the role of the source node in S_{n-1} , and the same skill can be applied recursively until the broadcasting is completed. Since the time required for generating the relay tree in S_n is $O(\log n)$, and the same scheme can be applied recursively in $S_n, S_{n-1}, S_{n-2}, \dots, S_2$, the time complexity for broadcasting in S_n is $O(n \log n)$. And this broadcasting algorithm is optimal in the one-port model [9].

For convenient description, we assume that the source node is A . It is clear that the relay tree is not unique, because the transmission sequence in phase 1 is not unique. Furthermore, it is not necessary that the recursive doubling scheme is applied. Thus, we shall propose another transmission sequence of phase 1 such that the number of link polarity changes is reduced. Since the method is recursive, we only consider how the message is sent from the source node in S_n to the $n-1$ leader nodes (i.e. the first iteration). Note that the root of the relay tree is the source node A . If the root is removed, the relay tree is split into $\log(n-1)$ relay subtrees. Assume $n-1 = 2^k$. The i -th subtree, $1 \leq i \leq k-1$, denoted as T_i , contains 2^{i-1} nodes. Each tree node is associated with a unique symbol in $W = \{i | 1 \leq i \leq n-1, i \neq A[1]\}$. W is partitioned into two subsets $W_1 = \{i | 1 \leq i \leq A[1]\}$ and $W_2 = \{i | A[1] < i \leq n-1\}$. Next, we further partition W_1 and W_2 into $\log(n-1)$ subsets by applying rule (R1), which will be given later. Each subset corresponds to the leftmost symbols in a subset of some subtree nodes.

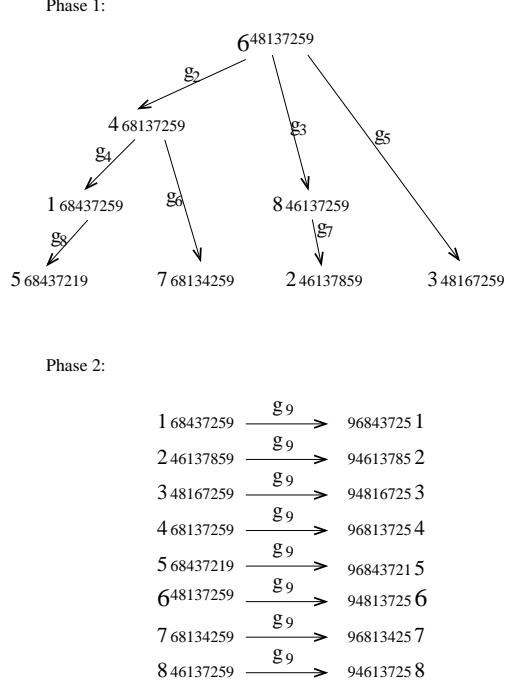


Figure 3: Mendia et al.'s optimal broadcasting in S_9 .

Before presenting our algorithm, we give the following lemma.

Lemma 1 *If $x = y + z$, where x, y and z are all positive integers, then there exist two sets M_1 and M_2 , such that $y \leq \sum_{i \in M_1} 2^i$, $z \leq \sum_{i \in M_2} 2^i$, where $M_1 \cap M_2 = \emptyset$ and $M_1 \cup M_2 = \{i | 0 \leq i \leq \lfloor \log_2 x \rfloor\}$.*

Proof: We first consider the case that $x = 2^{k+1} - 1$. Assume $x = 2^0 + 2^1 + 2^2 + \dots + 2^k$, where $k = \lfloor \log_2 x \rfloor$. Let $M = \{i | 0 \leq i \leq \lfloor \log_2 x \rfloor\}$. For any two arbitrary positive integers y and z , $y + z = x$, we can easily find a subset $M_1 \subset M$ such that $y = \sum_{i \in M_1} 2^i$. Let $M_2 = M - M_1$. Since $z = x - y$, it is obvious that $z = \sum_{i \in M_2} 2^i$.

Next, we consider the case that $x \neq 2^q - 1$ for any q . We can find x' , such that $x \leq x' = 2^0 + 2^1 + 2^2 + \dots + 2^k$. Suppose that $x = y + z$. We can find y' and z' such that $x' = y' + z' = (y + \Delta y) + (z + \Delta z)$, $\Delta y \geq 0$ and $\Delta z \geq 0$. By the first part of this proof, there exist M_1 and M_2 such that we can easily find a subset $M_1 \subset M$ such that $y' = \sum_{i \in M_1} 2^i$, $z' = \sum_{i \in M_2} 2^i$, $M_1 \cap M_2 = \emptyset$ and $M_1 \cup M_2 = M$. This completes the proof. \square

For example, $15 = 2^3 + 2^2 + 2^1 + 2^0$ can be partitioned into 9 and 6, and 9 can be expressed as $2^3 + 2^0$ and 6 can be expressed as $2^2 + 2^1$. 15 can also be partitioned into 10 and 5, and $10 = 2^3 + 2^1$ and $5 = 2^2 + 2^0$.

Take 11 as another example. $11 \leq 15 = 2^3 + 2^2 + 2^1 + 2^0$. We can use only $2^3, 2^2, 2^1$ and 2^0 to cover

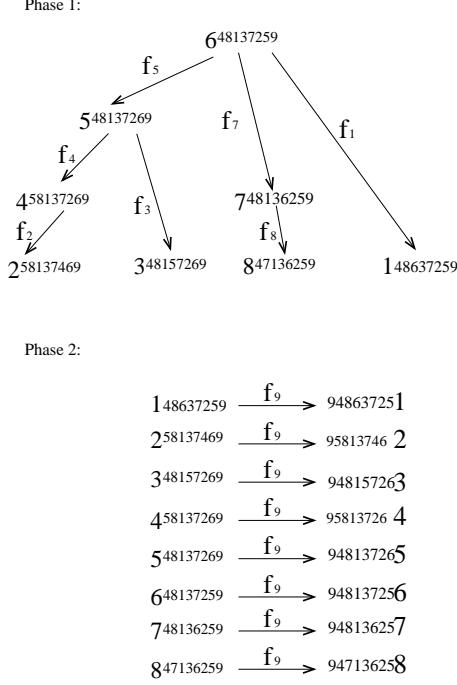


Figure 4: Broadcasting based on virtual channels in S_9 .

5 and 6, which partition 11. $5 = 2^2 + 2^0$ and $6 = 2^2 + 2^1 \leq 2^3 + 2^1$. Thus 5 is covered by 2^2 and 2^0 . And 6 is covered by 2^3 and 2^1 . If we partition 11 into 7 and 4, then $7 = 2^2 + 2^1 + 2^0$ and $4 = 2^2 \leq 2^3$. Thus 7 is covered by 2^2 , 2^1 and 2^0 and 4 is covered by 2^3 .

Applying Lemma 1, we can associate the symbols in W_1 and W_2 with some T_i 's as follows.

(R1) In each subset of W_1 and W_2 , symbols are consecutive. By Lemma 1, each subset can be covered by some 2^i 's, and thus these relay subtree T_i 's are associated with it.

For example , Figure 4 shows the broadcasting in S_9 . Suppose the source node is 648137259, then the symbols in $W_1 = \{1, 2, 3, 4, 5\}$ can be divided into $\{1\}$ and $\{2, 3, 4, 5\}$ and the symbols in $W_2 = \{7, 8\}$ forms only one subset $\{7, 8\}$. Then $\{1\}$, $\{7, 8\}$ and $\{2, 3, 4, 5\}$ corresponds to the relay subtrees T_1 , T_2 , and T_3 respectively. Note, in each relay subtree, the leftmost symbols of all nodes are either larger or smaller than $A[1]$.

In each subset, we try to select a symbol as the leftmost symbol of the root of the relay subtree. The root node is responsible for sending the message to other nodes in the relay subtree. The rule for selecting the leftmost symbol of the root is as follows.

(R2) If the symbols in the subset are all larger than

$A[1]$, then select the smallest symbol as the leftmost symbol of the root of the relay subtree; otherwise, select the largest symbol as the leftmost symbol of the root of relay subtree.

To avoid deadlock and provide adaptive capability, our *virtual channel allocation method* is as follows.

(1) Initially, the message is transmitted via the first virtual channel.

(2) Suppose that the message is routing in the i th virtual channel now. For a node, only if the polarity of the previous link is negative and the polarity of the next link to be routed is positive, then the message transmission is switched to the $(i+1)$ th virtual channel of the next link. In other cases, the message is still transmitted via the i th virtual channel.

Now, the 2-phase algorithm for broadcasting is modified and summarized as follows.

Phase 1: For the source node A , it sends message to the root of relay subtree T_i at step i . For the root of the relay subtree, if it receives the message, it sends the message to its child nodes by applying the recursive doubling scheme.

Phase 2: Each of the $n - 1$ nodes in the relay subtree, by applying $f_{A[n]}$, sends the message to the leader node in a substar S_{n-1} .

In S_n , the number of virtual channels for each physical link is $\lfloor \frac{n+1}{2} \rfloor$, which will be proved by the following theorem.

Theorem 1 *In our broadcasting approach, $\lfloor \frac{n+1}{2} \rfloor$ virtual channels are enough for each physical link in S_n .*

Proof: Clearly, the link polarity of the broadcast sequence in the relay tree will not change. The link polarity may change only when the message is sending from the relay nodes to the leader nodes of substars. Then the number of link polarity changes is at most $n - 1$ since the relay trees are recursively generated $n - 1$ times. Therefore, it needs at most $\lfloor \frac{n-1}{2} \rfloor + 1 = \lfloor \frac{n+1}{2} \rfloor$ virtual channels for each physical link. \square

It may cause deadlock when many nodes want to broadcast at the same time in the interconnection network. And our method can overcome this problem. To explain that our algorithm is deadlock-free, we need the following lemma.

Lemma 2 [12] *In a directed circuit on a star graph, there exists at least one link whose polarity is positive and at least one link whose polarity is negative.*

If some of the broadcasting paths form a circuit, then there must exist a physical link l , whose polarity

is positive and whose previous link polarity is negative. In our method, if the message is transmitted via virtual channel i of the previous link, then it will be transmitted via the virtual channel $i + 1$ of link l . Therefore, it will not cause a deadlock.

4 Broadcasting in the C_{n-1} Graph

In this section, we shall discuss broadcasting on the C_{n-1} graph. In our method, the virtual-channels needed for each physical link is $\lfloor \frac{n+1}{2} \rfloor$, as broadcasting in the complete star graph.

At the beginning, we briefly describe the broadcasting algorithm in $C_{n-1}(k)$ [8]. In phase 1, the source node A sends the message to the relay nodes. In phase 2, each of the relay nodes applies g_n (i.e. $f_{A[n]}$) to transmit the message to the leader node of one substar S_{n-1}^i , $n - k + 1 \leq i \leq n$. Note that the substar S_{n-1}^i , $1 \leq i \leq n - k$ does not exist. After each leader node receives the message, since each S_{n-1} is a complete star graph of dimension $n - 1$, the leader can be viewed as the source node in S_{n-1} , and the same skill in the complete star graph can be applied recursively until the broadcasting is completed.

Thus, the broadcasting algorithm in $C_{n-1}(k)$ with two phases of the first iteration can be summarized as follows.

Phase 1: By the recursive doubling scheme, a relay tree is generated such that each symbol in $\{A[1], A[2], \dots, A[n - 1]\}$ appears in the first symbols of the tree nodes exactly once. These tree nodes are called *relay nodes*.

Phase 2: For a relay node U in the relay tree, if $U[1] \geq n - k + 1$, then by applying $f_{U[n]}$, U sends the message to the leader node in substar $S_{n-1}^{U[1]}$.

For example, we consider the broadcasting in $C_8(7)$, in which the nodes are of the forms $xxxxxx9$, $xxx\dots x8$, \dots , $xxx\dots x4$ and $xxx\dots x3$. Assume the source node is $A = 648137259$, which will broadcast message to all other nodes in $C_8(7)$. Figure 5 shows this example. After the phase 1, nodes 148637259, 258137469, 348157269, 458137269, 548137269 648137259, 748136259, and 847136259 receive the message. In phase 2, node 348157269 sends message to node 948157263 which serves as the leader node of substar $xxx\dots x3$, i.e. S_8^3 , node 458137269 sends message to node 958137264 as the leader node of the substar $xxx\dots x4$, i.e. S_8^4 , \dots , node 847136259 sends message to node 947136258 as the leader node of the substar $xxx\dots x8$. i.e. S_8^8 . However, since substars $xxx\dots x1$ and $xxx\dots x2$ do not exist, nodes

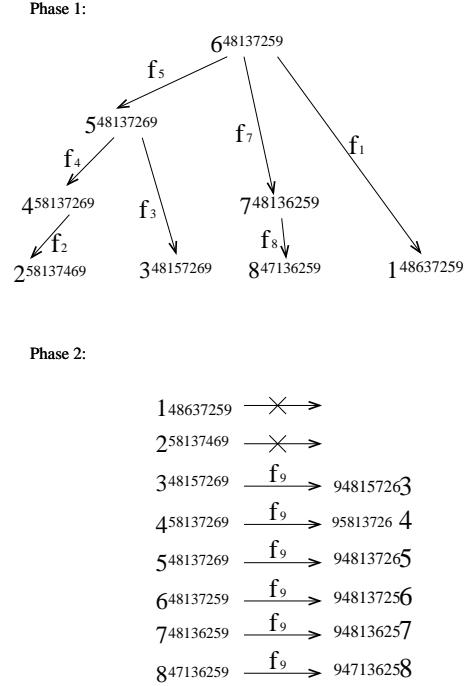


Figure 5: Broadcasting based on virtual channels in $C_8(7)$.

123456789 and 213456789 stop and will not send message to other nodes in phase 2. Now, each leader node which has received the message serves as the source of its substar and continues broadcasting the message to other nodes in its substar by recursively applying the broadcasting method in a complete star, mentioned in the previous section.

Theorem 2 *In our broadcasting approach, $\lfloor \frac{n+1}{2} \rfloor$ virtual channels are enough for each physical link in the C_{n-1} graph.*

Proof: In phase 1, C_{n-1} and S_n do the same work. The only difference in broadcasting in C_{n-1} and S_n is that in phase 2, not every node in C_{n-1} needs to send message to one leader node. So, the number of link polarity changes is still at most $n - 1$. Therefore, it needs at most $\lfloor \frac{n-1}{2} \rfloor + 1 = \lfloor \frac{n+1}{2} \rfloor$ virtual channels for each physical link. \square

5 Broadcasting without Message Redundancy in S_n

The broadcasting algorithm without message redundancy was proposed by Sheu et al. [10]. This algorithm consists of three phases. In phase 1, there

are $n - 2$ relay nodes in $S^A[n]_{n-1}$ which can receive the message from the source node A . As soon as the source node or relay nodes complete its phase 1 operation, it can apply phase 2 to send message to the leader of one of the $n - 1$ substars S_{n-1} except $S^{A[n]}_{n-1}$. After operations in phase 1 and phase 2 are completed, only the relay nodes continue to apply phase 3. The source node completes all of its operations and stops. In phase 3, the relay nodes in $S^{A[n]}_{n-1}$ should send the message to the other nodes in $S^{A[n]}_{n-1}$ and avoiding sending the message to a node which has received the message before in $S^{A[n]}_{n-1}$. $S^{A[n]}_{n-1}$ is decomposed into $n - 2$ substars S_{n-2} , $n - 3$ substars $S_{n-3}, \dots, n - i$ substars S_{n-i}, \dots, S_1 , and the source node. The $n - 2$ substars S_{n-2} , $n - 3$ substars S_{n-3}, \dots , and one S_1 are denoted as SG_{n-1} , SG_{n-2}, \dots, SG_3 , and SG_2 , respectively. Now, each of the $n - 2$ relay nodes can be viewed as the source node of one SG_i , $2 \leq i \leq n - 1$, and independently performs their phase 3 to broadcast the message within SG_i . SG_i has $i - 1$ S_{i-1} 's in it. One relay node is responsible for broadcasting message in these $i - 1$ substar S_{i-1} .

When the algorithm performs the phase 1 and phase 2, the relay tree in the previous section is used. We summary the 3-phase algorithm as follows.

Phase 1: The source node A sends the message to the root of relay subtree T_i at step i . After the root of T_i receives the message, it sends the message to its child nodes by applying the recursive doubling scheme.

Phase 2: By applying $f_{A[n]}$, the $n - 1$ relay nodes of the relay tree send the message to the leader node in each substar S_{n-1} .

Phase 3: The relay nodes, which receive the message in phase 1, recursively apply phase 1 and phase 2 to broadcast the message in SG_i .

For example, Figure 6 and Figure 7 show how the broadcasting is done in S_9 without message redundancy. Suppose the source node is 123456789. In phase 1, node 648137259 sends message to relay nodes 148637259, 258137469, 348157269, 458137269, 548137269 648137259, 748136259, and 847136259. In phase 2, the relay nodes perform $f_{A[9]}$ operation, sending message to the leader of other S_8 's. Now, the node 648137259 has completed its operations and stops. After doing the $f_{A[9]}$ operation, the relay nodes start their operations in phase 3. Node 847136259 serves as the source node of SG_8 (seven substar S_7 's), and broadcast message to other six S_7 's ($xx7xxxxx9$, $xx6xxxxx9$, $xx5xxxxx9$, $xx4xxxxx9$, $xx3xxxxx9$, $xx2xxxxx9$ and $xx1xxxxx9$). Node 348157269 broadcasts the message in SG_7 , node

Table 1: The decomposition of $xxxxxxxx9$ in S_9 in phase 3.

$S_8 : xxxxxxxx9$			
source in SG_i	SG_i	size	substars
847136259	SG_8	7 S_7 's	$xx1xxxxx9$ $xx2xxxxx9$ $xx3xxxxx9$ $xx4xxxxx9$ $xx5xxxxx9$ $xx6xxxxx9$ $xx7xxxxx9$
348157269	SG_7	6 S_6 's	$xx8x1xxx9$ $xx8x2xxx9$ $xx8x4xxx9$ $xx8x5xxx9$ $xx8x6xxx9$ $xx8x7xxx9$
258137469	SG_6	5 S_5 's	$xx8x3x1x9$ $xx8x3x4x9$ $xx8x3x5x9$ $xx8x3x6x9$ $xx8x3x7x9$
148637259	SG_5	4 S_4 's	$xx843x2x9$ $xx853x2x9$ $xx863x2x9$ $xx873x2x9$
748136259	SG_4	3 S_3 's	$xx81342x9$ $xx81352x9$ $xx81362x9$
458137269	SG_3	2 S_2 's	$x581372x9$ $x681372x9$
548137269	SG_2	1 S_1	$x48137269$
source node 648137259			

258137469 broadcasts the message in SG_6 , and so on. Table 1 shows the decomposition of the substar S_8 which the source node belongs to.

The number of virtual channels for each physical link is $\lfloor \frac{n+1}{2} \rfloor$ and proved in the following theorem.

Theorem 3 *In broadcasting without message redundancy, $\lfloor \frac{n+1}{2} \rfloor$ virtual channels are enough for each physical link in S_n .*

Proof: The number of polarity changes in phase 1 and phase 2 is the same as that we stated in Theorem 1. And the phase 3 reduces the steps for broadcasting message in the SG_i . Therefore, there are at most $\lfloor \frac{n-1}{2} \rfloor + 1 = \lfloor \frac{n+1}{2} \rfloor$ virtual channels required for each physical link. \square

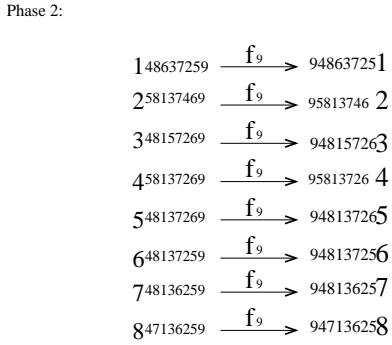
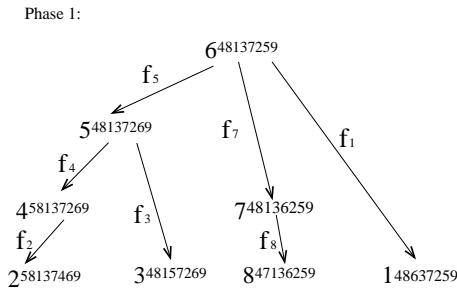


Figure 6: Phase 1 and phase 2 in the broadcasting on S_9 without message redundancy.

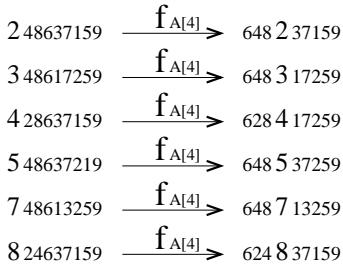
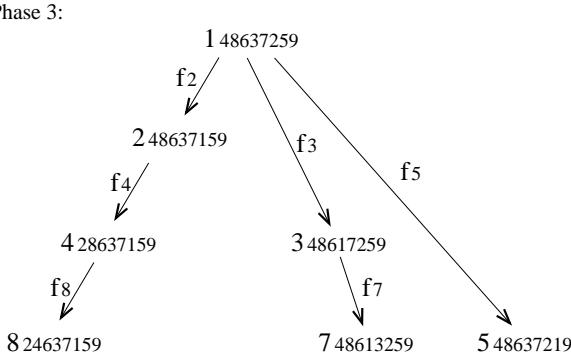


Figure 7: Phase 3 of the broadcasting on S_9 without message redundancy. Node $A = 648137259$ is the source node.

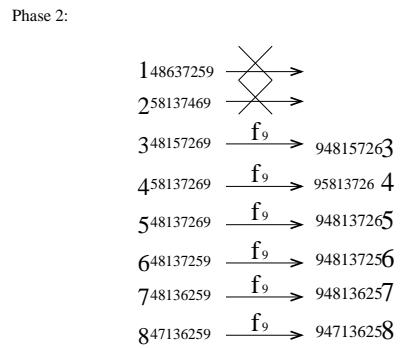
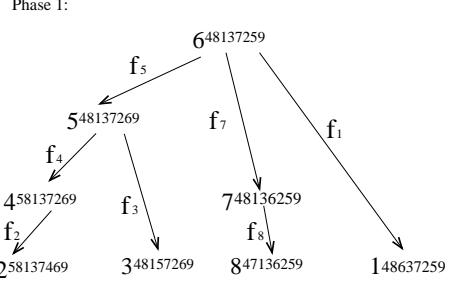


Figure 8: Phase 1 and phase 2 of broadcasting on the C_{n-1} graph without message redundancy.

6 Broadcasting without message redundancy in the C_{n-1} Graph

Based on the previous discussion, we can also perform the three-phase broadcasting algorithm in the C_{n-1} graph except that phase 2 is modified as follows.

Phase 2: For a relay node U in the relay tree, if $U[1] \geq n - k + 1$, then by applying $f_{U[n]}$, U sends the message to the leader node in substar $S_{n-1}^{U[1]}$.

For example, Figure 8 and Figure 9 show an example for broadcasting in $C_8(7)$ without message redundancy. The nodes in the C_{n-1} graph are of the format $xxxxxxxx9$, $xxxxxxxx8$, $xxxxxxxx7$, $xxxxxxxx6$, $xxxxxxxx5$, $xxxxxxxx4$ and $xxxxxxxx3$, which are the combination of 7 substars S_6 . The source node is $A = 648137259$, and it wants to broadcast message in the $C_8(7)$ graph.

First, in phase 1, node A sends message to relay nodes 148637259, 258137469, 348157269, 458137269, 548137269, 648137259, 748136259 and 847136259 in the relay tree. Then, in phase 2, only node 458137269, 548137269, 748136259, 847136259 and 648137259 need to perform $f_{A[9]}$ to send the message to nodes 958137264, 948137265, 948137256, 948136257 and 947136258, which are the source nodes of the substars $xxxxxxxx4$, $xxxxxxxx5$, $xxxxxxxx6$, $xxxxxxxx7$,

Phase 3:

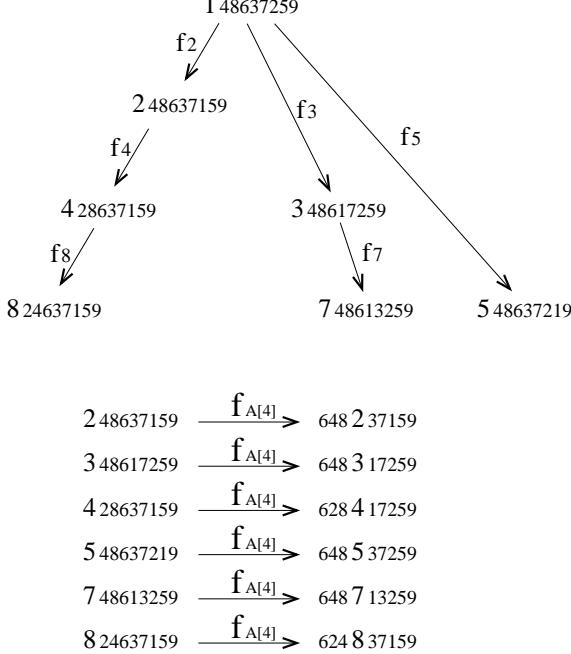


Figure 9: Phase 3 of broadcasting on the C_{n-1} graph without message redundancy. Node $A = 648137259$ is the source node.

and $xxxxxx8$, respectively.

Then, we illustrate the operations in phase 3 . All the relay nodes, which received the message in phase 1, start their phase 3 except the source node 2134567.

Now, we decompose the substar S_8 ($xxxxxx8$) into SG_8 (7 S_7 's), SG_7 (6 S_6 's), ..., SG_1 (1 S_1 's) and the source node in S_9 . Every relay node receiving the message in phase 1 serves as the source of an SG . In particular, we choose the node 847136259 to illustrate the procedure. In phase 3, node 847136259 serves as the source node of 7 substars S_7 ($xx7xxxxx9$, $xx6xxxxx9$, $xx5xxxxx9$, $xx4xxxxx9$, $xx3xxxxx9$, $xx2xxxxx9$ and $xx1xxxxx9$). Node 847136259 is in substar $xx7xxxxx9$, so it needs only to send message to other 6 substars S_7 . By the same skill we use in phase 1, node 847136259 sends message to nodes 147836259, 247136859, 347186259, 478136259, 548137269, 648137259 and 748136259. Then, these seven nodes apply $f_A[3]$ to send the message to nodes 741836259, 742136859, 743186259, 874136259, 845137269, 846137259, and 847136259, respectively.

Similarly, the number of virtual channels for each physical link for broadcasting without message redundancy in the C_{n-1} graph is still $\lfloor \frac{n+1}{2} \rfloor$.

7 Conclusion

Processors often need to communicate with each other by sending and receiving messages in a multiprocessors system. The main objectives in designing communication algorithms are high speed and low contention. High-speed communication algorithms must use a low-latency communication technique such as virtual cut-through. To avoid deadlock in broadcasting, we use the concept of virtual channel to develop our broadcasting algorithms on the star graph. A new relation between two nodes , f , also used here. We construct a new broadcasting tree, the relay tree, to reduce the number of virtual channels required for each physical link by reducing the number of polarity change.

References

- [1] S. B. Akers, D. Horel, and B. Krishnamurthy, “The star graph: An attractive alternative to the n-cube,” *Proceeding of the International Conference on Parallel Processing*, pp. 393–400, 1987.
- [2] S. B. Akers and B. Krishnamurthy, “A group-theoretical model for symmetric interconnection networks,” *IEEE Transactions on Computers*, Vol. 38, No. 4, pp. 555–566, 1989.
- [3] B. Alspach, “Cayley graphs with optimal fault tolerance,” *IEEE Transactions on Computers*, Vol. 41, No. 10, pp. 1337–1339, 1992.
- [4] W. J. Dally, “Virtual-channel flow control,” *IEEE Transactions on Parallel and Distributed Systems*, Vol. 3, No. 2, pp. 194–205, Mar. 1992.
- [5] W. J. Dally and H. Aoki, “Deadlock-free adaptive routing in multicomputer networks using virtual channels,” *IEEE Transactions on Parallel and Distributed Systems*, pp. 466–475, Apr. 1993.
- [6] K. Day and A. Tripathi, “A comparative study of topological properties of hypercubes and star graphs,” *IEEE Transactions on Parallel and Distributed Systems*, Vol. 5, No. 1, pp. 31–38, Jan. 1994.
- [7] S.-C. Hu and C.-B. Yang, “Fault tolerance on star graphs,” *International Journal of Foundations of Computer Science*, Vol. 8, No. 2, pp. 127–142, 1997.

- [8] S. Latifi and N. Bagherzadeh, “Incomplete star: an incrementally scalable network based on the star graph,” *IEEE Transactions on Parallel and Distributed Systems*, Vol. 5, No. 1, pp. 97–102, Jan. 1994.
- [9] V. E. Mendaia and D. Sarker, “Optimal broadcasting on the star graph,” *IEEE Transactions on Computers*, Vol. 3, No. 4, pp. 389–396, Jan. 1992.
- [10] J. P. Sheu, C. T. Wu, and T. S. Chen, “An optimal broadcasting algorithm without message redundancy in star graphs,” *IEEE Transactions on Parallel and Distributed Systems*, Vol. 6, No. 6, pp. 653–658, 1995.
- [11] S. Sur and P. K. Srimani, “A fault tolerant routing algorithm in star graph interconnection networks,” *Proceeding of the International Conference on Parallel Processing*, Vol. 3, pp. 267–270, 1991.
- [12] C.-B. Yang and T.-H. Liu, “Wormhole routing on the star graph interconnection network,” *Proc. of the 4th Australian Theory Symposium, CAT’98*, pp. 51–65, Feb. 1998.