故障のあるスターネットワーク上の最適なブロードキャスティング

梅 傲寒　鮑 豊　濱田 幸弘　五十嵐 善英

群馬大学 工学部 情報工学科

我々は、故障のあるスターネットワークにおいて一つブロードキャスティングアルゴリズムを提案し、ネットワーク上に高々 $n-2$ 個の故障が生じた場合に、そのアルゴリズムは $O(n\log n)$ の時間でブロードキャスティングを終了することができることを示す。

Optimal Time Broadcasting in Faulty Star Networks

(Extended Abstract)

Aohan Mei, Feng Bao, Yukihiro Hamada, and Yoshihide Igarashi
{mei, bao, hamada, igarashi}@comp.cs.gunma-u.ac.jp

We propose a non-adaptive single-port broadcasting scheme in the n-star network such that it tolerates $n-2$ faults even in the worst case and completes the broadcasting in $O(n\log n)$ time. The existence of such a broadcasting scheme was not known before. The technique used in the broadcasting scheme is called ***diffusing-and-disseminating***. This technique is useful to overcome various difficulties for the fault tolerance of broadcasting in star networks. We also analyze the reliability of the broadcasting scheme in the case where faults are randomly distributed in the n-star network.

1 Introduction

Broadcasting is one of the fundamental tasks in network communications. It is the process of disseminating a message from the source node to all other nodes in the network. It can be accomplished in such a way that each node repeatedly receives and forwards messages. For the past decade an overwhelming amount of studies on broadcasting in networks have been done. There are some good survey papers on this subject, e.g.,[8],[16].

Star networks were proposed as attractive interconnection networks [1]. In recent years, star networks have been much studied [2],[3],[5],[10],[13],[15], and a lot of results on broadcasting in star networks have been derived [4],[6],[7],[10],[14]. Star networks have recursive structures. The n-star network consists of n $(n-1)$-star networks and additional $n!/2$ links. The connectivity and the diameter of the n-star network are $n-1$ and $\lfloor 3(n-1)/2 \rfloor$, respectively [2],[1]. As for broadcasting in star networks the following results are known: (1) There exists a single-port broadcasting scheme with running time at most $n\log n$ in the n-star with no faults [9]. The scheme exploits the recursive structure of star networks. It is optimal in the sense that for any constant $c<1$, there does not exist any single-port broadcasting scheme with running time $cn\log n$. This is because $2^{cn\log n} < n!$ for any constant $c<1$. (2) For the case of a faulty n-star network, there exists a single-port broadcasting scheme with running time $O(n^2)$. The broadcasting by the scheme tolerates $n-2$ faults even in the worst case. The principle of the fault-tolerant broadcasting is the same as that in hypercubes proposed in [12]. We will explain this broadcasting scheme in Section 3, and it will be used in the second stage of the broadcasting proposed in this paper. (3) A multi-port fault-tolerant broadcasting using routing was introduced in [4], and its running time is $O(n^{\frac{3}{2}}\log n)$. Gargano et al. gave a better multi-port fault-tolerant broadcasting scheme, and its running time is $O(n)$ [7]. Mendia and Sarker presented a problem of finding a single-port broadcasting scheme in faulty star networks in [9]. However, any substantial solution to this problem has not been given. We give an optimal solution to this problem in this paper.

2 Preliminaries

Let $a_1a_2\cdots a_n$ be a permutation of n symbols $1,2,\cdots,n$. For an integer $2\le i\le n$ and a permutation $a_1a_2\cdots a_n$, a generator g_i is defined as $g_i(a_1a_2\cdots a_n)=a_ia_2\cdots a_{i-1}a_1a_{i+1}\cdots a_n$. An undirected graph $G=(V,E)$ is called the n-star graph (denoted by S_n) if $V=\{a_1a_2\cdots a_n \mid a_1a_2\cdots a_n$ is a permutation of $1, 2, \cdots, n\}$ and $E = \{(u, v) \mid u, v \in V$ and $v = g_i(u)$ for some $i\}$. The n-star graph is also called the n-star network. For $2 \leq i \leq n$, edges of S_n specified by g_i are said to be of dimension i. We often denote edge (u, v) by g_i^u if $g_i(u) = v$. It is immediate that S_n has $n!$ nodes and it is $(n-1)$-regular. We can choose $(n-1)!$ permutations with an identical last symbol from $n!$ permutations. Hence, we can decompose the n-star network into n node-disjoint $(n-1)$-star networks. From this property we can say that star networks are *hierarchical.*

Let $a_1, a_2, \cdots, a_m$ be m distinct symbols chosen from $\{1, 2, \cdots, n\}$. There are $m!$ permutations of $a_1, a_2, \cdots, a_m$. For each permutation $a_1 a_2 \cdots a_m$, we assign an integer from range $[1, 2, \cdots, m!]$ in the lexicographic order. This integer is called the rank of $a_1 a_2 \cdots a_m$ and denoted by $r(a_1 a_2 \cdots a_m)$.

In Section 4, we propose a fault-tolerant broadcasting scheme that exploits smaller sub-star networks S_i's $(2 \leq i < n)$ recursively. In the analysis of the scheme, we specify sub-star networks in the following way. Each sub-star network S_i of S_n is specified by a permutation of n symbols and the set of generators $\{g_2, g_3, \cdots, g_i\}$. In other words, for each i, S_n can be decomposed into $(n!)/(i!)$ disjoint S_i's by partitioning the node set so that nodes belong to the same part of the decomposition if and only if $n-i$ symbols from the rightmost of all the nodes of the part are an identical sequence. Note that each sub-star network S_i can be also decomposed into disjoint smaller sub-star networks. For simplicity, we use $1 * n$ to denote a permutation with the first and last symbols being 1 and n, respectively. This notation can be extended to denote other permutations. For example, $1 * n - 1n$ denotes a permutation with the first, the $(n-1)$st and the nth symbols being 1, $n-1$ and n, respectively.

We assume that each node represents a processor and each link represents a bidirectional communication line connecting two nodes at the extremes of the link. All the nodes in a network are synchronized with a global clock. Broadcasting time is measured as the number of steps to complete the broadcasting. In each step every node can send a message to at most one neighbor node and can receive a message from at most one neighbor node. If more than one neighbor nodes attempt to send messages to the same node in the same step, the node can receive just one message from any one of the neighbor nodes. Such a model is called a single-port network. Communication from node $s_1 s_2 \cdots s_n$ to node $a_1 a_2 \cdots a_n$ can be done in the same fashion as communication from the identity permutation $12 \cdots n$ to node $f(a_1) f(a_2) \cdots f(a_n)$, where $f(s_k) = k$ for each $k (1 \leq k \leq n)$. Hence, it is sufficient to consider broadcasting only from the source node $12 \cdots n$. All logarithms in this paper are to the base 2.

3 An Information Disseminating Scheme

In this section we give a natural broadcasting scheme in S_n. The principle of the broadcasting scheme is the same as that of broadcasting in hypercubes given in [7] and [19]. This scheme can be implemented in the single-port manner. It is described as the following procedure. The procedure is executed at each node u of S_n concurrently.

```
procedure    Dissem(n, t)
   (* it is executed at each node u *)
   repeat t times
      for i := 2 to n do
         if u held the message before the current step then
            u sends the message along dimension i
```

For the implementation of procedure *Dissem*(n, t), it is not necessarily that each node knows whether its incident nodes and links are healthy and which is the source node of broadcasting. The following lemma is immediate.

Lemma 3.1 *Let s be the source node in S_n and $p(s, u)$ be a path of length t from s to node u. Then the message from s will reach u through $p(s, u)$ by executing procedure* Dissem(n, t) *at each node in S_n, provided there are no faults on $p(s, u)$.*

Theorem 3.2 *If there are at most $n-2$ faulty nodes and/or links in S_n, then every healthy node receives the message from the source node by executing procedure* Dissem$(n, \lfloor 3(n-1)/2 \rfloor + 4)$ *at each node in S_n.*

Corollary 3.3 *If there are at most $n-2$ faulty nodes and/or links in S_n, then the message from the source node can reach every node within distance d by executing procedure* Dissem$(n, d+4)$ *at each node in S_n.*

The next theorem shows that $\Theta(n^2)$ steps are necessary to complete broadcasting in S_n by procedure *Dissem.*

Theorem 3.4 *For any $t < \lfloor n/2 \rfloor$, procedure* Dissem(n, t) *cannot complete broadcasting even if there are no faults in S_n.*

From Theorem 3.3 and Theorem 3.4 we can say that procedure *Dissem* can broadcast a message from the source node throughout the network S_n in $3n^2/2 + O(n)$ steps if there are at most $n - 2$ faults, but cannot complete broadcasting in S_n in $\lfloor n/2 \rfloor (n-1)$ steps even if there are no faults.

4 An Efficient Fault-Tolerant Broadcasting Scheme

We now describe our fault-tolerant broadcasting scheme. Let d be the minimum positive integer such that $d! \geq n - 1$. Let $p = p_1 p_2 \cdots p_n$ be an arbitrary permutation of $1, 2, \cdots, n$, and let $p[i, j]$ denote $p_i p_{i+1} \cdots p_j$, where $1 \leq i \leq j \leq n$. We partition the label of each permutation p into three intervals. The first interval is just the first symbol $p[1, 1]$ and called the *head.* The second interval is $p[2, 3d+1]$ and called the *identifier district.* The identifier district is divided into three blocks, $p[2, d+1]$, $p[d+2, 2d+1]$, and $p[2d+2, 3d+1]$. The last interval is $p[3d+2, n]$. (See Figure 1.)

The broadcasting process by our scheme is divided into two stages, called the *diffusing* stage and the *disseminating* stage. In the *diffusing* stage, the message from the source node $s = 12 \cdots n$ is transmitted along $n-1$ internally disjoint channels. Each channel contains at least one node in every $(n-3d-1)$-sub-star network. Note that there are totally $\frac{n!}{(3d+1)!}$ such sub-star networks. In other words, after the *diffusing* stage, at least one node in each $(n - 3d - 1)$-sub-star network holds the message if there exist at most $n - 2$ faults in S_n. Hence, after the *diffusing* stage, for every node u in S_n, there exists a node v holding the message within distance $\lfloor \frac{9d}{2} \rfloor$ from u. Then, $Dissem(n, \lfloor \frac{9d}{2} \rfloor + 4)$ is executed in the *disseminating* stage. By Corollary 3.3, every node can receive the message in this way if there exist at most $n-2$ faults in S_n. The *diffusing* stage consists of the *pre-stage* and the *recursive stage.* During the *pre-stage,* s sends

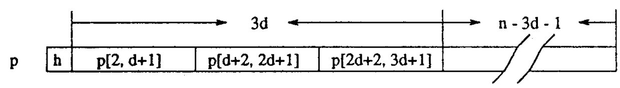

Figure 1: The identifier district of p.

its message to its $n-2$ neighbors $g_2(s) = 213 \cdots n$, $g_3(s) = 3214 \cdots n$, ..., $g_{n-1}(s) = (n-1)2 \cdots (n-2)1n$. Then the $n - 1$ nodes with the message (including s) send the message along dimension n. Then, $n - 1$ nodes $t_1 = n * 1$, $t_2 = n * 2$, ... , $t_{n-1} = n * (n-1)$ receive the message. Next for each j $(1 \leq j \leq n-1)$, the message is transmitted from t_j to w_j along an appropriate route, where w_j satisfies the conditions that $w_j[3d+2, n] = t_j[3d+2, n]$ and $r(w_j[2, d+1]) = r(w_j[d+2, 2d+1]) = r(w_j[2d+2, 3d+1]) = j$. For each j $(1 \leq j \leq n-1))$ we can specify such a route from t_j to w_j by choosing appropriate nodes whose labels are obtained by some changes of symbols in $t_j[2, d+1]$, $t_j[d+2, 2d+1]$, and $t_j[2d+2, 3d+1]$. For any pair of distinct i and j $(1 \leq i, j \leq n-1)$ the route from t_i to w_i and the route from t_j to w_j are node-disjoint since the last symbol of the label of each node on the former route is i while the last symbol of each node on the latter route is j.

The *recursive* stage follows the *pre-stage.* The *recursive* stage is consistent with the hierarchical structure of S_n. For clear explanation, we now assume that no faults exist in S_n. Suppose that there exist $n - 1$ nodes holding the message in an m-sub-star network in the recursion stage. Let these $n - 1$ nodes be $x_1, x_2, \cdots, x_{n-1}$, where for each j $(1 \leq j \leq n-1)$, $r(x_j[2, d+1]) = r(x_j[d+2, 2d+1]) = r(x_j[2d+2, 3d+1]) = j$. Then for each j $(1 \leq j \leq n-1)$, the message is transmitted from x_j to m modes that are in different $(m-1)$-sub-star networks. That is, for each j $(1 \leq j \leq n-1)$, x_j transmits the message to a node, say y_j in each of the $m-1$ S_{m-1}, where y_j does not necessarily satisfy $r(y_j[2, d+1]) = r(y_j[d+2, 2d+1]) = r(y_j[2d+2, 3d+1]) = j$. However, at least two of $r(y_j[2, d+1])$, $r(y_j[d+2, 2d+1])$ and $r(y_j[2d+2, 3d+1])$ are equal to j. If one of these three ranks is not equal to the others, then we can choose a route from y_j to a node in the same S_{m-1}, say z_j which is obtained by some changes of symbols in the corresponding intervals so that $r(z_j[2, d+1]) = r(z_j[d+2, 2d+1]) = r(z_j[2d+2, 3d+1]) = j$. Then we move to the next round of the recursive stage. The recursive stage stops when it reaches $(3d+1)$-sub-star networks.

4.1 Construction of $n-1$ Node-Disjoint i-Level Channels

Definition 1 *For a node p in S_n, the identifier (id for short) of p is $r(p[i,i+d])$ if $r(p[i,i+d]) = r(p[j,j+d])$ for some $i \neq j$ $(i,j = 1, d+1$ or $2d+1)$, and otherwise it is 0.*

Lemma 4.1 *For any pair of permutations p_1 and p_2 of length n, they are distinct if their id's are not identical*

We now explain the process of the recursive stage in detail. For each j $(1 \leq j \leq n-1)$, w_j has the message at the beginning of the recursive stage. We therefore may consider that each w_j is a message source at the beginning of the recursive stage. Hereafter, we call them message sources. Remember that $r(w_j[2,d+1]) = r(w_j[d+2,2d+1]) = r(w_j[2d+2,3d+1]) = j$. The recursive stage is divided into $n-3d-1$ rounds. For each i $(1 \leq i \leq n-3d-1)$ and each j $(1 \leq j \leq n-1)$, let $W_i(w_j)$ be the set of nodes that hold the message from w_j and are ready to broadcast the message at the beginning of round i. Initially, let $W_1(w_j) = \{w_j\}$ and $W_i(w_j) = \phi$ $(i > 1)$. Its contents will be renewed at the beginning of each round i. For each i $(1 \leq i \leq n-3d-1)$ and each j $(1 \leq j \leq n-1)$, during round i each node in $W_i(w_j)$ executes the following operations:

(1) Each node $p \in W_i(w_j)$ sends its message to $g_2(p), g_3(p), \cdots, g_{3d+1}(p)$ sequentially. Then node p broadcasts its message in a binary jumping way as described in Section 4.2. That is, p sends its message to $g_{3d+1+1}(p), g_{3d+1+2}(p), \cdots, g_{3d+1+2^k}(p)$ sequentially, where k is the maximum integer such that $3d+1+2^k \leq n-i$.

When all of $g_2(p), g_3(p), \cdots, g_{3d+1}(p)$ have received the message, each node $u \in \{g_2(p), g_3(p), \cdots, g_{3d+1}(p)\}$ executes the following operations:

(2) Along g_{n-i+1}, u sends the message received from p. Then after $g_{n-i+1}(u)$ receives the message, it transmits the message to node u' through a path such that $r(u'[2,d+1]) = r(u'[d+2,2d+1]) = r(u'[2d+2,3d+1]) = j$. Let $Route_p^{(i,j)}(u)$ be the set of nodes on the path from u to u'. Node u' is added to $W_{i+1}(w_j)$.

Let $3d+1 < dim \leq 3d+1+2^k$. Each node v in the set of nodes (including $g_{3d+1+1}(p), g_{3d+1+2}, \cdots, g_{3d+1+2^k}(p)$) that received the message along an edge g_{dim}^v broadcasts the message in a binary jumping way as described in (3) below:

(3) When p sends a message along $g^p_{3d+1+2^{l-1}}$, v first receives a message from g^v_{dim} (it is not necessarily that v receives the message directly from p). Then v sends its message to $g_{dim+2^l}(v), g_{dim+2^{l+1}}(v), \cdots, g_{dim+2^q}(v)$ sequentially, where q is the maximum integer such that $dim + 2^q \leq n-i$. When the binary jumping transmissions finished, p and all the nodes that received the message along a dimension among $g_{3d+2}, g_{3d+3}, \cdots, g_{n-i}$, send the message along g_{n-i+1}. The set of the final $n-3d-i$ nodes that have received the message along g_{n-i+1} is denoted by $BJ_i(p)$. All the nodes in $BJ_i(p)$ are added to $W_{i+1}(w_j)$, too.

The operations listed (1), (2) and (3) above are called Rule (1), Rule (2) and Rule (3), respectively. We will give a further detailed description about our broadcasting scheme in Subsection 4.2.

Lemma 4.2 *Let $w_j = *j$ $(1 \leq j < n)$ be a message source in S_n at the beginning of the recursive stage. For each i $(1 \leq i \leq n-3d)$ and for an arbitrary node $p \in W_i(w_j)$, $r(p[2,d+1]) = r(p[d+2,2d+1]) = r(p[2d+2,3d+1]) = j$, where $1 \leq i \leq n-3d$.*

Definition 2 *Let $p \in W^i(w_j)$, $H = \{g_2(p), g_3(p), \cdots, g_{3d+1}(p)\}$ and $Y_i(w_j,p) = \bigcup_{u \in H} Route_p^{(i,j)}(u)$. Let $X_i(w_j,p)$ be the set of nodes that have received the message directly or indirectly from node p but not in $Y_i(w_j,p)$ during round i. We define $U_i(w_j) = \bigcup_{p \in W_i(w_j)} Y_i(w_j,p)$ and $V_i(w_j) = \bigcup_{p \in W_i(w_j)} X_i(w_j,p)$.*

Definition 3 *Let w_j be a message source in S_n at the beginning of the recursive stage. For $1 \leq i < n-3d$, the i-*level channel *rooted at w_j, denoted by $C_i(w_j)$, is defined as $\bigcup_{k=1}^{i} (U_k(w_j) \cup V_k(w_j))$.*

The next lemma is immediate from the definition of the *i-level channel*.

Lemma 4.3 *For $1 \leq i < n-3d$, at least one node in each S_{n-i} obtained by fixing the last i symbols of the labels of nodes of S_n is contained in the i-level channel rooted at w_j, $C_i(w_j)$.* □

Lemma 4.4 *Let $w_j = *j$ $(1 \leq j < n)$ be a message source in S_n at the beginning of the recursive stage satisfying $r(w_j[2, d+1]) = r(w_j[d+2, 2d+1]) = r(w_j[2d+2, 3d+1]) = j$. Then for each i $(1 \leq i < n-3d)$ there exists the i-level channel rooted at w_j such that the id of every node in $\bigcup_{k=1}^{n-3d-1} U_k(w_j)$ is j.*

Proof. By the definition of the i-level channel rooted at w_j, $C_i(w_j) = \bigcup_{k=1}^{i} (U_k(w_j) \cup V_k(w_j))$. From Lemma 4.2, for each node $p \in W_i$, $r(p[2, d+1]) = r(p[d+2, 2d+1]) = r(p[2d+2, 3d+1]) = j$. According to Rule 3 of the recursive stage, the identifier district of each node in $V_i(w_j)$ $(1 \leq i < n-3d)$ remains unchanged. Hence, the *id*'s of all the nodes in $\bigcup_{k=1}^{n-3d-1} V_k(w_j)$ are equal to j. On the other hand, from Rule 1 and Rule 2 of the recursive stage, at most one of $u[2, d+1]$, $u[d+2, 2d+1]$ and $u[2d+2, 3d+1]$ of each node $u \in U_i(w_j)$ $(1 \leq i < n-3d)$ has changed. Hence, the *id*'s of all the nodes in $\bigcup_{k=1}^{n-3d-1} U_k(w_j)$ are equal to j. □

From Lemma 4.1 and Lemma 4.4, the following theorem is immediate.

Theorem 4.5 *Let $w_1, w_2, \cdots, w_{n-1}$ be $n-1$ message sources in S_n. Then for each i $(1 \leq i < n-3d)$, there exist $n-1$ node-disjoint i-level channels $C_i(w_1), C_i(w_2), \cdots, C_i(w_{n-1})$.* □

Lemma 4.6 *For $1 \leq i < n-3d$, the time needed in round i of the recursive stage is $3d + \max\{\log(n - 3d - i) + 1, \lfloor \frac{3d}{2} \rfloor + 1\}$.*

Theorem 4.7 *Let s be the source node in S_n. For an arbitrary sub-star network S_{3d+1} obtained by fixing the last $n-3d-1$ symbols of the labels of nodes in S_n, s can send the message to $n-1$ distinct nodes in the S_{3d+1} through $n-1$ node-disjoint paths within $n + \lfloor \frac{9d}{2} \rfloor - 1 + \sum_{i=1}^{n-3d-1} (3d + \max\{\log(n-3d-i)+1, \lfloor \frac{3d}{2} \rfloor + 1\})$ steps.*

4.2 Diffuse-Disseminate Scheme in Faulty Star Networks

We are now ready to formally describe our fault tolerant broadcasting scheme in S_n. We assume that there exist at most f faults in S_n, where $f < n-1$. Our broadcasting scheme, called the *Diffuse-Disseminate*, consists of two stages. This scheme is described as follows:

```
procedure  Diffuse-Disseminate
(* for each node u *)
if u is the source node then /* the pre-stage */
   for i := 2 to n − 1 do
      send the message along g_i^u
if u has the message then send the message along g_n^u
if u received a message from g_n^u then
   for i := 1 to ⌊9d/2⌋ do
      adjust u so that r(u[2, d+1]) = r(u[d+2, 2d+1]) = r(2d+2, 3d+1]) = u_n
      (* u_n denotes the last symbol of u *)
for i := 0 to n − 3d − 2 do /* the recursive stage */
begin
   if u has the message then
   begin
      for j := 2 to 3d + 1 do
         send the message along g_j^u
      call Binary-jump(3d + 1, n − i)
   end
   if u received the message from g_j^u (1 < j < 3d + 2) then send the message along g_{n-i}^u
   if u received a message from g_{n-i}^u then call Route(u)
end
call Dissem(n, ⌊9d/2⌋ + 4) /* the disseminating stage */
```

Binary-jump(n_1, n_2) is to distribute the message from a node in S_n to its neighbors in a binary jumping way. *Route*(u) can transmit a message from u to a node with no destroyed block in its *id* district. Here, we omitted their descriptions.

Theorem 4.8 *Procedure* Diffusing-and-Disseminating *can broadcast a message from the source node to all other nodes in* S_n *within* $(1+\varepsilon)n\log n$ *steps if there exist at most* $n-2$ *faulty nodes and/or links in the network, where* ε *is a positive constant less than 1.*

5 Analysis of Broadcasting in S_n with Random Faults

Since the connectivity of S_n is $n-1$, any broadcasting scheme in S_n can tolerate at most $n-2$ faults. However, if we assume that faulty places are randomly distributed in a network then the worst case occurs rarely [5]. Hence, even if there exist much more than $n-2$ faults in S_n, broadcasting may succeed with a high probability. In this section, we give a probabilistic analysis of the reliability of our broadcasting scheme and have the following theorem.

Theorem 5.1 *For any constant* $\alpha < 1$, *if there are no more than* $(n!)^\alpha$ *faulty nodes randomly distributed in* S_n, *broadcasting by our scheme succeeds with a probability higher than* $1-1/n!$.

6 Conclusion

We showed that our broadcasting scheme tolerates up to $n-2$ faults in S_n and that its running time is $O(n\log n)$. This running time is optimal for the asymptotic order and almost optimal for the constant factor of the order. We conjecture that the scheme might be optimal even for the constant factor of the order, too. This problem is theoretically interesting, and worthy for the further investigation.

References

[1] S. B. Akers, D. Harel, and B. Krishnamurthy, "The star graph: An attractive alternative to the n-cube." In *Proc. Int. Conf. Parallel Processing,* pp. 393–400, 1987.

[2] S. B. Akers and B. Krishnamurthy, "On group graphs and their fault tolerance." *IEEE Trans. Computers,* Vol. C-36, pp. 885–888, 1987.

[3] S. B. Akers and B. Krishnamurthy, "A group-theoretic model for symmetric interconnection networks." *IEEE Trans. Computers,* Vol. 38, pp. 555–566, 1989.

[4] N. Bagherzadeh, N. Nassif, and S. Latifi, "A routing and broadcasting scheme on faulty star graphs." *IEEE Trans. Computers,* Vol. 42, pp. 1398–1403, 1993.

[5] K. Day and A. Tripathi, "A comparative study of topological properties of hypercubes and star graphs." *IEEE Trans. Parallel and Distributed Systems,* Vol. 5, pp. 31–38, 1994.

[6] P. Fragopoulou and S. G. Akl, "Optimal communication algorithms on star graphs using spanning tree constructions." *Journal of Parallel and Distributed Computing,* Vol. 24, pp. 55–71, 1995.

[7] L. Gargano, A. A. Rescigno, and U. Vaccaro, "Optimal communication in faulty star networks." Manuscript, 1995.

[8] S. M. Hedetniemi, S. T. Hedetniemi, and A. L. Liestman, "A survey of gossiping and broadcasting in communication networks." *Networks,* Vol. 18, pp. 319–349, 1988.

[9] V. E. Mendia and D. Sarkar, "Optimal broadcasting on the star graph." *IEEE Trans. Parallel and Distributed Systems,* Vol. 3, pp. 389–396, 1992.

[10] A. Menn and A. K. Somani, "An efficient sorting algorithm for the star graph interconnection network." In *Proc. Int. Conf. Parallel Processing,* Vol. III, pp. 1–8, 1990.

[11] A. Pelc, "Fault-tolerant broadcasting and gossiping in communication networks." manuscript, 1995.

[12] P. Ramanathan and K. G. Shin, "Reliable broadcast in hypercube multicomputers." *IEEE Trans. Computers,* Vol. 37, pp. 1654–1657, 1988.

[13] Y. Rouskov and P. K. Srimani, "Fault diameter of star graphs." *Information Processing Letters,* Vol. 48, pp. 243–251, 1993.

[14] J.-P. Sheu, W.-H. Liaw, and T.-S. Chen, "A broadcasting algorithm in star graph interconnection networks." *Information Processing Letters,* Vol. 48, pp. 237–241, 1993.

[15] S. Sur and P. K. Srimani, "A fault tolerant algorithm in star graph interconnection networks." In *Proc. Int. Conf. Parallel Processing,* Vol. III, pp. 267–270, 1991.